Behind the Tech with Kevin Scott

EP 02 – Anders Hejlsberg: A craftsman of computer language

1

00:00:01,070 --> 00:00:04,750

>> Surely. Anything

can be done here.

2

00:00:04,750 --> 00:00:06,830

Look at this diagram. I

just drew it for you.

3

00:00:06,830 --> 00:00:08,135

Just go and make it so, right?

4

00:00:08,135 --> 00:00:10,950

No, coding is hard and

it continues to be hard,

5

00:00:10,950 --> 00:00:12,540

and code gets bigger

and bigger and bigger

6

00:00:12,540 --> 00:00:14,350

but our brains are not

getting any bigger,

7

00:00:14,350 --> 00:00:18,100

and this is largely

a brain exercise.

8

00:00:23,440 --> 00:00:26,240

>> Hi everyone. Welcome

to Behind the Tech.

9

00:00:26,240 --> 00:00:27,510

I'm your host, Kevin Scott,

10

00:00:27,510 --> 00:00:29,715

Chief Technology

Officer for Microsoft.

11

00:00:29,715 --> 00:00:32,045

In this podcast, we're going

to get Behind the Tech.

12

00:00:32,045 --> 00:00:33,420
We'll talk with
some of the people

13
00:00:33,420 --> 00:00:34,680
who made our modern tech world

14
00:00:34,680 --> 00:00:36,030
possible and understand what

15
00:00:36,030 --> 00:00:38,115
motivated them to
create what they did.

16
00:00:38,115 --> 00:00:39,570
So join me to

17
00:00:39,570 --> 00:00:41,130
maybe learn a little bit
about the history of

18

00:00:41,130 --> 00:00:44,280
computing and get
a few behind the scenes insights

19
00:00:44,280 --> 00:00:45,525
into what's happening today.

20
00:00:45,525 --> 00:00:51,380
Stick around.

21
00:00:51,380 --> 00:00:52,460
>> In this episode
of Behind the Tech,

22
00:00:52,460 --> 00:00:54,140
we'll meet Anders Hejlsberg.

23
00:00:54,140 --> 00:00:57,545
Anders has always been
one of my coding heroes.

24

00:00:57,545 --> 00:01:00,155
He built Turbo Pascal,

25
00:01:00,155 --> 00:01:02,660
at Borland, which
is the tool that

26
00:01:02,660 --> 00:01:05,430
I use to become a real
software engineer.

27
00:01:05,430 --> 00:01:08,035
At Borland, he had a long career

28
00:01:08,035 --> 00:01:10,190
where he was Chief Architect of

29
00:01:10,190 --> 00:01:12,785
Delphi in addition to
the Turbo Pascal tools

30

00:01:12,785 --> 00:01:14,990
and eventually moved
over to Microsoft,

31
00:01:14,990 --> 00:01:19,410
where he helped create C# as
the Lead Language Designer,

32
00:01:19,410 --> 00:01:20,850
and today, he spends

33
00:01:20,850 --> 00:01:23,235
his time as the Core
Developer on TypeScript.

34
00:01:23,235 --> 00:01:25,580
Anders has had a 35-year

35
00:01:25,580 --> 00:01:28,760
career building development tools

36

00:01:28,760 --> 00:01:30,780

that software developers love.

37

00:01:30,780 --> 00:01:32,570

We're going to learn how he's

38

00:01:32,570 --> 00:01:35,550

done that today on

Behind the Tech.

39

00:01:36,190 --> 00:01:39,530

>> So, welcome, and thanks for

40

00:01:39,530 --> 00:01:42,905

being my first guest

on this debut podcast.

41

00:01:42,905 --> 00:01:44,830

>> Thanks for being

willing to experiment.

42

00:01:44,830 --> 00:01:47,910

>> You were a calculated first

choice because,

43

00:01:47,910 --> 00:01:49,430

I think I've told

you this before,

44

00:01:49,430 --> 00:01:53,090

but I learned to program

when I was 12 years old.

45

00:01:53,090 --> 00:01:57,060

My entry was BASIC in

6502 Assemby language,

46

00:01:57,060 --> 00:01:58,250

the 80 Assembly language.

47

00:01:58,250 --> 00:02:00,110

The first real programming

48

00:02:00,110 --> 00:02:02,555

I ever did was on Turbo Pascal.

49

00:02:02,555 --> 00:02:06,815

I went to a science and

technology high school and took

50

00:02:06,815 --> 00:02:11,395

Intro to CS class and Turbo

Pascal 5.5 was the tool.

51

00:02:11,395 --> 00:02:14,510

I don't think I would

have chosen a career in

52

00:02:14,510 --> 00:02:16,070

computer science

if it hadn't been

53

00:02:16,070 --> 00:02:17,650

for that sort of

confluence of things.

54

00:02:17,650 --> 00:02:21,155

So like in a way, you're

responsible for my career.

55

00:02:21,155 --> 00:02:21,890

>> That's awesome.

56

00:02:21,890 --> 00:02:24,020

>> The main reason I wanted

to have you on the show

57

00:02:24,020 --> 00:02:27,200

is this must be the case for

58

00:02:27,200 --> 00:02:29,495

a huge number of folks

59

00:02:29,495 --> 00:02:33,455

and we'll get into

your journey as an engineer.

60

00:02:33,455 --> 00:02:36,140

But what made you

decide that you wanted

61

00:02:36,140 --> 00:02:38,810

to build programming

and development tools?

62

00:02:38,810 --> 00:02:40,610

Because that's sort of

been your entire career.

63

00:02:40,610 --> 00:02:41,870

>> Yes, it has. You

know I've been doing it

64

00:02:41,870 --> 00:02:43,330

for more than 35 years now,

65

00:02:43,330 --> 00:02:45,295
and it's kind of
scary to think back

66
00:02:45,295 --> 00:02:47,550
and it's such a
long time ago now.

67
00:02:47,550 --> 00:02:49,445
I mean, and the world
and the industry,

68
00:02:49,445 --> 00:02:51,220
there wasn't really
an industry even.

69
00:02:51,220 --> 00:02:52,530
IT was so different.

70
00:02:52,530 --> 00:02:53,750
I mean, in the industry, I got

71

00:02:53,750 --> 00:02:55,340

started probably

in the early '80s,

72

00:02:55,340 --> 00:02:57,200

but I got started coding in

73

00:02:57,200 --> 00:02:59,890

high school back

in the late '70s.

74

00:02:59,890 --> 00:03:04,305

Probably I was trying to

think back '78 or, yeah, '78.

75

00:03:04,305 --> 00:03:06,020

>> Were you the type

of tinkerer where

76

00:03:06,020 --> 00:03:08,015

you were building a little

programming language?

77

00:03:08,015 --> 00:03:10,939

>> No, no. So I was born

and grew up in Denmark,

78

00:03:10,939 --> 00:03:13,400

and I went to a high school

outside of Copenhagen.

79

00:03:13,400 --> 00:03:15,950

It was one of

the first high schools

80

00:03:15,950 --> 00:03:19,000

to offer students

access to a computer.

81

00:03:19,000 --> 00:03:20,155

>> Yeah. What was the computer?

82

00:03:20,155 --> 00:03:23,520

>> There was the old HP

2100 it was called.

83

00:03:23,520 --> 00:03:26,200

It had 32K of

ferrite core memory.

84

00:03:26,200 --> 00:03:28,095

You can literally open it and see

85

00:03:28,095 --> 00:03:30,445

the ferrite core.

So it was amazing.

86

00:03:30,445 --> 00:03:34,450

Paper tape reader and

then after a while,

87

00:03:34,450 --> 00:03:38,910

they got a 14-inch one

megabyte hard drive,

88

00:03:38,910 --> 00:03:41,525

which was absolutely

state of the art.

89

00:03:41,525 --> 00:03:42,650

>> It was like huge.

90

00:03:42,650 --> 00:03:44,810

>> It was enormous. At

the read-write head

91

00:03:44,810 --> 00:03:47,230

on that thing had

this gigantic magnet,

92

00:03:47,230 --> 00:03:49,580

and everything was so primitive.

93

00:03:49,580 --> 00:03:50,990

You could program it, and we put

94

00:03:50,990 --> 00:03:53,330

that poor computer

through so much torture,

95

00:03:53,330 --> 00:03:55,100

like trying to make

music by moving

96

00:03:55,100 --> 00:03:57,455

the read-write head

on the hard drive.

97

00:03:57,455 --> 00:04:00,650

They would vibrate

the whole tape, and it was-.

98

00:04:00,650 --> 00:04:02,000

>> Yeah. Yeah.

99

00:04:02,000 --> 00:04:03,170

>> So that was sort of a very

100

00:04:03,170 --> 00:04:05,980

hands-on introduction

to computing,

101

00:04:05,980 --> 00:04:08,150

and I too remember that you could

102

00:04:08,150 --> 00:04:10,580

either program it

in Assembly code.

103

00:04:10,580 --> 00:04:12,575

I think there was

a FORTRAN compiler

104

00:04:12,575 --> 00:04:14,940

but then there was

an HP ALGOL compiler.

105

00:04:14,940 --> 00:04:15,710

>> Oh, Interesting.

106

00:04:15,710 --> 00:04:18,350

>> That was actually

sort of how I vectored

107

00:04:18,350 --> 00:04:21,740

into that branch of programming

languages ultimately.

108

00:04:21,740 --> 00:04:23,990

Because that was the first

language I was taught.

109

00:04:23,990 --> 00:04:25,210

We didn't have BASIC.

110

00:04:25,210 --> 00:04:27,015

There was no BASIC

on that computer.

111

00:04:27,015 --> 00:04:27,645

Do you know what I mean?

112

00:04:27,645 --> 00:04:29,330

So I learned ALGOL and this

113

00:04:29,330 --> 00:04:32,295

was a very primitive compiler.

114

00:04:32,295 --> 00:04:33,780

I mean, it didn't

support recursion.

115

00:04:33,780 --> 00:04:34,220

>> Right.

116

00:04:34,220 --> 00:04:38,830

>> That machine had no stacks

so call instructions would

117

00:04:38,830 --> 00:04:41,030
just store the return address
and the first word and

118
00:04:41,030 --> 00:04:43,550
then return was an indirect
jump back to that.

119
00:04:43,550 --> 00:04:44,570
Right? I'm probably going to get

120
00:04:44,570 --> 00:04:45,620
in trouble for saying it,

121
00:04:45,620 --> 00:04:47,870
but better than FORTRAN or COBOL.

122
00:04:47,870 --> 00:04:49,235
>> They were still
better than FORTRAN.

123

00:04:49,235 --> 00:04:51,550

>> Yeah. Yeah. But anyway-.

124

00:04:51,550 --> 00:04:53,360

>> How much were you

all influenced by

125

00:04:53,360 --> 00:04:55,890

the Scandinavian

programming languages?

126

00:04:55,890 --> 00:04:58,520

So Niklaus Wirth

was the designer of

127

00:04:58,520 --> 00:05:01,945

the Pascal programming language

and then MODULO later on.

128

00:05:01,945 --> 00:05:03,090

>> It's funny, in high school, I

129

00:05:03,090 --> 00:05:04,390

didn't know anything

about any of that,

130

00:05:04,390 --> 00:05:06,860

and really, it was just

sort of learning by doing.

131

00:05:06,860 --> 00:05:09,350

Then, I started

electrical engineering

132

00:05:09,350 --> 00:05:11,735

at the Danish

engineering academy.

133

00:05:11,735 --> 00:05:13,535

There, I met a guy

134

00:05:13,535 --> 00:05:15,320

who had just transferred

from another school

135

00:05:15,320 --> 00:05:17,450

into the engineering academy

136

00:05:17,450 --> 00:05:19,510

and we both liked playing cards,

137

00:05:19,510 --> 00:05:20,980

and he lost some money.

138

00:05:20,980 --> 00:05:24,560

So I had some IOUs so I

had to get to know him.

139

00:05:24,560 --> 00:05:26,930

So I ended up getting

involved with

140

00:05:26,930 --> 00:05:29,585

him and starting a computer

company in Copenhagen.

141

00:05:29,585 --> 00:05:32,165

We had the very

first retail store

142

00:05:32,165 --> 00:05:34,790

where you could go in

and buy a kit computer.

143

00:05:34,790 --> 00:05:35,200

>> Wow.

144

00:05:35,200 --> 00:05:37,265

>> We were right at

the cusp at that time

145

00:05:37,265 --> 00:05:40,745

of microprocessors

becoming democratized.

146

00:05:40,745 --> 00:05:43,245

The 8080 and the Z-80.

147

00:05:43,245 --> 00:05:44,810

I'm like this is

when the Sinclair

148

00:05:44,810 --> 00:05:47,020

ZX80 started to happen.

149

00:05:47,020 --> 00:05:49,850

There was this British kid

computer called the NASSCOMM,

150

00:05:49,850 --> 00:05:51,670

which had a Z-80 in it,

151

00:05:51,670 --> 00:05:53,800

but I started writing

software for that one then.

152

00:05:53,800 --> 00:05:57,510
It came actually with
a Microsoft ROM BASIC.

153
00:05:57,510 --> 00:05:59,450
>> So what sort of
software were you writing?

154
00:05:59,450 --> 00:06:01,830
>> Well, so first, I
started writing games.

155
00:06:01,830 --> 00:06:03,280
It's funny, I never liked playing

156
00:06:03,280 --> 00:06:04,480
games but I liked writing

157
00:06:04,480 --> 00:06:07,540
games. >>Were these sort of
text-based adventure games?

158

00:06:07,540 --> 00:06:07,890

>> Yeah.

159

00:06:07,890 --> 00:06:09,480

>> It was like Moon Lander and

160

00:06:09,480 --> 00:06:11,570

Star Trek and that sort

of thing, right?

161

00:06:11,570 --> 00:06:13,200

You could actually

buy these books.

162

00:06:13,200 --> 00:06:15,060

I think it was like

Byte Magazine had

163

00:06:15,060 --> 00:06:16,720

these 101 computer games

164

00:06:16,720 --> 00:06:18,805

where you could type

it all in, right?

165

00:06:18,805 --> 00:06:21,580

>> Just great. Like it

was such a great thing.

166

00:06:21,580 --> 00:06:22,730

>> It was awesome.

So I learned a lot.

167

00:06:22,730 --> 00:06:24,040

But then, I started actually

168

00:06:24,040 --> 00:06:25,450

getting into assembler coding and

169

00:06:25,450 --> 00:06:26,860

sort of trying to

understand more about

170

00:06:26,860 --> 00:06:28,420

how the machine worked.

171

00:06:28,420 --> 00:06:29,785

Then I got curious about

172

00:06:29,785 --> 00:06:31,670

extending the

Microsoft ROM BASIC.

173

00:06:31,670 --> 00:06:33,130

Because that ROM BASIC,

174

00:06:33,130 --> 00:06:35,180

it had to fit into an 8K ROM.

175

00:06:35,180 --> 00:06:37,105

So there were a bunch of

commands it didn't have.

176

00:06:37,105 --> 00:06:38,575

There was no renumber command,

177

00:06:38,575 --> 00:06:39,990

which was a royal pain

in the neck,

178

00:06:39,990 --> 00:06:41,370

because if you ran out num now,

179

00:06:41,370 --> 00:06:44,490

you had to manually

go retype every line.

180

00:06:44,490 --> 00:06:46,435

But there were

some extension points

181

00:06:46,435 --> 00:06:47,950

where you could actually

sort of hook into

182

00:06:47,950 --> 00:06:49,450

the ROM BASIC and

because they were

183

00:06:49,450 --> 00:06:51,940

an extra slot for E prompts

on the motherboard.

184

00:06:51,940 --> 00:06:55,790

I wrote this little 4K

ROM BASIC extension

185

00:06:55,790 --> 00:06:58,350

that gave you renumbered and

a bunch of other things.

186

00:06:58,350 --> 00:07:01,100

So that was like a little

plug-in tool kit. Right?

187

00:07:01,100 --> 00:07:03,680

>> How old were you then? This

is like after university?

188

00:07:03,680 --> 00:07:06,210

>> Yeah, I was probably 20.

189

00:07:06,210 --> 00:07:08,350

>> So like very, very early in

190

00:07:08,350 --> 00:07:09,550

your career you were

191

00:07:09,550 --> 00:07:11,540

mucking around with

your programming tools.

192

00:07:11,540 --> 00:07:12,360

>> Sure. Yeah.

193

00:07:12,360 --> 00:07:16,030

>> Is that just sort of
a sensibility thing or-

194
00:07:16,030 --> 00:07:18,270
>> I just sort of
found it interesting.

195
00:07:18,270 --> 00:07:20,070
There are a lot of
challenges there, right?

196
00:07:20,070 --> 00:07:22,940
Then that interest further
got piqued by- now that

197
00:07:22,940 --> 00:07:24,995
I've figured out, "Oh, you
could actually the extend."

198
00:07:24,995 --> 00:07:27,150
I remember still missing

199

00:07:27,150 --> 00:07:28,720

that ALGOL programming language

200

00:07:28,720 --> 00:07:29,980

and I was telling my buddy,

201

00:07:29,980 --> 00:07:30,610

"Maybe I should write

202

00:07:30,610 --> 00:07:32,020

a little ALGOL thing that

we could probably-.".

203

00:07:32,020 --> 00:07:34,230

"No, no. There's

this new thing called Pascal.

204

00:07:34,230 --> 00:07:35,460

You really should

check that out.".

205

00:07:35,460 --> 00:07:37,530

"Of course. Okay. Okay".

Then I started like,

206

00:07:37,530 --> 00:07:38,910

"Oh, this is better. "

207

00:07:38,910 --> 00:07:40,950

Because one of

the things that Niklaus

208

00:07:40,950 --> 00:07:43,485

did consistently through

his entire career,

209

00:07:43,485 --> 00:07:45,290

he's the creator of Pascal,

210

00:07:45,290 --> 00:07:46,450

but he was involved with

211

00:07:46,450 --> 00:07:49,070

ALGOL and then later

MODULO and Oberon,

212

00:07:49,070 --> 00:07:51,995

and every language got

progressively simpler.

213

00:07:51,995 --> 00:07:52,515

>> Yes.

214

00:07:52,515 --> 00:07:55,270

>> ALGOL had all

these complex call by

215

00:07:55,270 --> 00:07:56,650

value and call by name and

216

00:07:56,650 --> 00:07:58,720

then Pascal dumped

a bunch of that,

217

00:07:58,720 --> 00:08:00,610

and then MODULO dumped

even more of that and

218

00:08:00,610 --> 00:08:02,500

that Oberon got even

more minimalistic.

219

00:08:02,500 --> 00:08:02,780

>> Yeah.

220

00:08:02,780 --> 00:08:04,150

>> I remember I've never written

221

00:08:04,150 --> 00:08:07,050

a real ALGOL program but

because I was a compiler guy,

222

00:08:07,050 --> 00:08:09,830

the ALGOL grammar was

all over the place.

223

00:08:09,830 --> 00:08:11,065

>> It was complicated.

224

00:08:11,065 --> 00:08:12,795

>> So complicated.

225

00:08:12,795 --> 00:08:14,435

>> So anyway, so

there, I set to work

226

00:08:14,435 --> 00:08:16,405

and I wrote this little Pascal

227

00:08:16,405 --> 00:08:17,695

that was effectively

228

00:08:17,695 --> 00:08:20,380

the nascent Turbo Pascal,

if you will, right?

229

00:08:20,380 --> 00:08:23,470

It had a little onscreen editor

and a runtime library

230

00:08:23,470 --> 00:08:26,715

and a compiler and it was all

squeezed into 12K in a ROM.

231

00:08:26,715 --> 00:08:28,675

>> What gave you

the idea to do that?

232

00:08:28,675 --> 00:08:30,615

Because that's sort of

a new thing, right?

233

00:08:30,615 --> 00:08:32,825

That might have been

the first ID, right?

234

00:08:32,825 --> 00:08:34,570

>> Yeah. That was probably one of

235

00:08:34,570 --> 00:08:37,165

the very first IDs ever. Yeah.

236

00:08:37,165 --> 00:08:38,005

>> That's amazing.

237

00:08:38,005 --> 00:08:41,360

>> Well, UCSD Pascal

was on the Apple too,

238

00:08:41,360 --> 00:08:42,770

but it was sort of different.

239

00:08:42,770 --> 00:08:45,280

It was a PCODE interpreter and

240

00:08:45,280 --> 00:08:49,270

the ID was not really

quite as user-friendly.

241

00:08:49,270 --> 00:08:53,995

It was more like a sort of

semi-command liney ID thing.

242

00:08:53,995 --> 00:08:56,380

>> So you wrote this thing

in mostly in Assembly?

243

00:08:56,380 --> 00:08:57,730

>> It was all in Z-80 Assembly.

244

00:08:57,730 --> 00:08:59,830

Yeah. Oh, yeah.

245

00:08:59,830 --> 00:09:02,155

Now, that was my weapon

of choice at the time,

246

00:09:02,155 --> 00:09:07,105

and that grew into a CP/M-80.

247

00:09:07,105 --> 00:09:10,510

It was obvious that one was

a subset implementation,

248

00:09:10,510 --> 00:09:13,090

and then that grew into

a fuller subset and then

249

00:09:13,090 --> 00:09:16,575

finally a full

implementation for CP/M-80.

250

00:09:16,575 --> 00:09:18,665

At that point, we met the guys.

251

00:09:18,665 --> 00:09:20,395

Borland was founded in Denmark.

252

00:09:20,395 --> 00:09:21,500

A lot of people don't know that.

253

00:09:21,500 --> 00:09:24,250

We knew the original founders

because they had had

254

00:09:24,250 --> 00:09:26,350

another computer company and

255

00:09:26,350 --> 00:09:28,330

it was a very small

industry in Denmark.

256

00:09:28,330 --> 00:09:31,750

They were writing some

stuff in Pascal MT Plus.

257

00:09:31,750 --> 00:09:35,045

I remember from

digital research that CP/M.

258

00:09:35,045 --> 00:09:36,505

That was a horrible Pascal.

259

00:09:36,505 --> 00:09:40,630

It was like one of those

insert-pass-one disk in drive.

260

00:09:40,630 --> 00:09:42,820

Right? Then they would

261

00:09:42,820 --> 00:09:44,440

grind and then you'd

take that out and

262

00:09:44,440 --> 00:09:45,230

then it would write some of

263

00:09:45,230 --> 00:09:46,270

the temporary code and then you'd

264

00:09:46,270 --> 00:09:48,760

insert the code generator disk.

265

00:09:48,760 --> 00:09:52,110

It took an afternoon to

compile a small program.

266

00:09:52,110 --> 00:09:54,670

There were like dialects of

Pascal that were just like

267

00:09:54,670 --> 00:09:57,540

not as nice as They

were Turbo Pascal.

268

00:09:57,540 --> 00:10:00,265

>> Right. Then they were

afraid of making extensions,

269

00:10:00,265 --> 00:10:01,870

and then, I was never

really afraid of

270

00:10:01,870 --> 00:10:03,880

making extensions if

they were useful.

271

00:10:03,880 --> 00:10:04,210

>> Yeah.

272

00:10:04,210 --> 00:10:05,650

>> So these guys were "Wait.

273

00:10:05,650 --> 00:10:07,060

We got actually

274

00:10:07,060 --> 00:10:08,650

this implementation you

should check it out".

275

00:10:08,650 --> 00:10:09,760

They didn't believe

a word of what we are

276

00:10:09,760 --> 00:10:11,345

saying because

that's not possible.

277

00:10:11,345 --> 00:10:13,505

>> I just want to double-click

on this point again.

278

00:10:13,505 --> 00:10:15,310

Coming up with one of

279

00:10:15,310 --> 00:10:16,900

the first integrated development

280

00:10:16,900 --> 00:10:18,030

environments that you have

281

00:10:18,030 --> 00:10:20,895

written in Z-80 Assembly

language at that point,

282

00:10:20,895 --> 00:10:23,575

that's a unbelievable

breakthrough.

283

00:10:23,575 --> 00:10:25,515

>> I suppose in retrospect, yes.

284

00:10:25,515 --> 00:10:26,770

I've never really thought of it

285

00:10:26,770 --> 00:10:27,955

that way. But, you know, it's-

286

00:10:27,955 --> 00:10:28,470

>> Just incredible. [inaudible].

287

00:10:28,470 --> 00:10:32,680

>> It just seemed

like this is going to

288

00:10:32,680 --> 00:10:34,420
be so much better
than having to have

289
00:10:34,420 --> 00:10:35,475
a first date a load

290
00:10:35,475 --> 00:10:36,920
an editor and then
load at the bargain.

291
00:10:36,920 --> 00:10:39,220
Why not just put it all
to there? I don't know.

292
00:10:39,220 --> 00:10:40,810
I never really- and especially,

293
00:10:40,810 --> 00:10:42,970
at the time, because
again, like more framing.

294

00:10:42,970 --> 00:10:45,060

These are not Windowed systems,

295

00:10:45,060 --> 00:10:47,410

can't have multiple things

opened at the same time.

296

00:10:47,410 --> 00:10:51,220

It's super tedious to switch

from one program to another.

297

00:10:51,220 --> 00:10:54,480

So like having everything in

one place is just as huge

298

00:10:54,480 --> 00:10:57,555

>> Totally. The added

compile, run, debug,

299

00:10:57,555 --> 00:11:01,480

cycle, just shrunk by

many orders of magnitude.

300

00:11:01,480 --> 00:11:03,630

>> Yeah, and I'm embarrassed

to say I've forgotten what

301

00:11:03,630 --> 00:11:05,845

was it F9 to compile and run,

302

00:11:05,845 --> 00:11:07,090

or was it F5?

303

00:11:07,090 --> 00:11:08,580

>> I don't even

remember what it was.

304

00:11:08,580 --> 00:11:10,410

I think it was F5 yeah but.

305

00:11:10,410 --> 00:11:11,400

>> It was like miraculous.

306

00:11:11,400 --> 00:11:13,940

>> Maybe F3 but

yeah, it was great.

307

00:11:13,940 --> 00:11:15,880

There were all sorts of tricks in

308

00:11:15,880 --> 00:11:17,820

there like the

runtime library was

309

00:11:17,820 --> 00:11:21,950

the first 12K of the system

and then when producing code,

310

00:11:21,950 --> 00:11:26,130

I just copy the first 12K

into the X we were producing.

311

00:11:26,130 --> 00:11:27,560

There is your runtime

library, right?

312

00:11:27,560 --> 00:11:31,130

Then generate code

from there on out and

313

00:11:31,130 --> 00:11:32,615

you could compile the memory

314

00:11:32,615 --> 00:11:34,675

and we'd put the code in

memory and run it, right?

315

00:11:34,675 --> 00:11:37,705

Or the original implementation

compile to tape,

316

00:11:37,705 --> 00:11:40,305

to floppy tape and then you go,

317

00:11:40,305 --> 00:11:42,810
sorry to the tape recorder
interface, right?

318
00:11:42,810 --> 00:11:45,440
Then you can load that
machine code up because

319
00:11:45,440 --> 00:11:48,190
I mean there was only 64K of
memory. I mean it was crazy.

320
00:11:48,190 --> 00:11:50,545
>> Yeah. So, I bought a copy of

321
00:11:50,545 --> 00:11:53,900
Turbo Pascal 5.5
out of a catalog called

322
00:11:53,900 --> 00:11:55,430
Programmers Paradise.

323

00:11:55,430 --> 00:11:58,165

This is just sort of how you

used to buy software and

324

00:11:58,165 --> 00:12:00,965

so I forked over my two

hundred bucks or whatever.

325

00:12:00,965 --> 00:12:02,095

>> Oh no, it wasn't even that.

326

00:12:02,095 --> 00:12:04,360

It was $49 like $49.95.

327

00:12:04,360 --> 00:12:06,190

>> It was affordable

because I was poor.

328

00:12:06,190 --> 00:12:07,965

So thank you for

making cheap software.

329

00:12:07,965 --> 00:12:08,870

>> Yeah.

330

00:12:08,870 --> 00:12:11,010

>> But like it came on

like this bundle of

331

00:12:11,010 --> 00:12:12,500

five and a quarter

inch floppy disk

332

00:12:12,500 --> 00:12:15,645

that sort of fed into

my and I was lucky...

333

00:12:15,645 --> 00:12:18,180

>> I mean I got to get credit

there to the Borland guys,

334

00:12:18,180 --> 00:12:20,420

to Felipe and Neil

and Mogens and all of

335

00:12:20,420 --> 00:12:22,760

the original founders of

Borland because what they did,

336

00:12:22,760 --> 00:12:26,390

they ended up licensing this

Pascal compiler that I had in

337

00:12:26,390 --> 00:12:29,420

my small company and we were

selling it for 500 bucks

338

00:12:29,420 --> 00:12:30,940

and doing business

the way business

339

00:12:30,940 --> 00:12:33,060

was normally done for

a software, right?

340

00:12:33,060 --> 00:12:35,080

Then they licensed

it. Of course we

341

00:12:35,080 --> 00:12:37,100

never thought to put

a minimum price on

342

00:12:37,100 --> 00:12:38,930

what they were supposed to

sell the product for and then

343

00:12:38,930 --> 00:12:40,955

come back with we're going

to sell it for $49.95.

344

00:12:40,955 --> 00:12:42,915

What? You got to.

345

00:12:42,915 --> 00:12:45,590

It's just terrible.

It's supposed to sell

346

00:12:45,590 --> 00:12:48,260

for 10 times. And so

they cut the price by

347

00:12:48,260 --> 00:12:51,650

10x and then they literally sold

348

00:12:51,650 --> 00:12:54,950

four or five orders of

magnitude more copies of it,

349

00:12:54,950 --> 00:12:56,380

right? That was tremendous.

350

00:12:56,380 --> 00:12:57,695

>> Yeah, I mean when

you think about

351

00:12:57,695 --> 00:12:59,470
the life cycle of innovation,

352
00:12:59,470 --> 00:13:02,075
it's like not always just
about the technical thing.

353
00:13:02,075 --> 00:13:04,735
Like that wouldn't have been
an appropriate environment

354
00:13:04,735 --> 00:13:07,270
for students if they
cost 500 bucks.

355
00:13:07,270 --> 00:13:09,290
>> At that time it
was like, "Hey,

356
00:13:09,290 --> 00:13:13,210
if you only charge
50 bucks, why even pirate?

357

00:13:13,210 --> 00:13:14,550

It was worth just to have

358

00:13:14,550 --> 00:13:16,670

the manual that actually tells

you how it works, right?

359

00:13:16,670 --> 00:13:16,895

>> Yeah.

360

00:13:16,895 --> 00:13:19,195

So one of the other things

that was really good about

361

00:13:19,195 --> 00:13:21,870

Turbo Pascal was the manuals

were actually quite good.

362

00:13:21,870 --> 00:13:24,520

How big a push did

you have in that?

363

00:13:24,520 --> 00:13:26,055

>> I had written manuals for

364

00:13:26,055 --> 00:13:28,190

the Pascal compiler that we

had in our company called

365

00:13:28,190 --> 00:13:29,955

Poly Pascal and it had

366

00:13:29,955 --> 00:13:32,510

a pretty extensive manual

for the language.

367

00:13:32,510 --> 00:13:35,355

The introductory parts were

not as great and Oler

368

00:13:35,355 --> 00:13:38,415

who wrote the documentation

Oler Rassmussen.

369

00:13:38,415 --> 00:13:40,325

He did a great job

on turning that

370

00:13:40,325 --> 00:13:42,230

into something that was really

371

00:13:42,230 --> 00:13:44,430

a fun book to read

and so a lot of

372

00:13:44,430 --> 00:13:47,140

people were very happy with

the documentation as well.

373

00:13:47,140 --> 00:13:48,890

So it was just a confluence

374

00:13:48,890 --> 00:13:51,790
of a bunch of factors
that just came

375
00:13:51,790 --> 00:13:54,105
together right. And the right
thing at the right time

376
00:13:54,105 --> 00:13:56,920
and the industry was so
nascent. I mean that was it.

377
00:13:56,920 --> 00:13:59,725
>> So going back to
your university days,

378
00:13:59,725 --> 00:14:01,945
did you have a professor

379
00:14:01,945 --> 00:14:05,700
or colleagues or like
someone who was really

380

00:14:05,700 --> 00:14:08,560

great at the programming

language the compilers part of

381

00:14:08,560 --> 00:14:10,275

the curriculum where they sort of

382

00:14:10,275 --> 00:14:12,555

lit this aha moment for you.

383

00:14:12,555 --> 00:14:15,045

>> Not a whole lot actually.

384

00:14:15,045 --> 00:14:16,595

I did electrical engineering

385

00:14:16,595 --> 00:14:18,675

because in order to

do computer science,

386

00:14:18,675 --> 00:14:19,935

you would have gone to

387

00:14:19,935 --> 00:14:22,330

the university. But I

was always more of

388

00:14:22,330 --> 00:14:24,400

an engineering sort of

focused and I wasn't

389

00:14:24,400 --> 00:14:25,995

necessarily sure that programming

390

00:14:25,995 --> 00:14:27,730

was what I wanted to do.

Do you know what I mean?

391

00:14:27,730 --> 00:14:28,535

>> Yeah.

392

00:14:28,535 --> 00:14:30,690

>> At the school at the time,

393

00:14:30,690 --> 00:14:34,340

they had yet to introduce

personal computers.

394

00:14:34,340 --> 00:14:38,260

So, they had the Northern

European Computing Center

395

00:14:38,260 --> 00:14:40,215

there where you would

turn in your stack of

396

00:14:40,215 --> 00:14:43,115

punch cards and then

the next day you'd get

397

00:14:43,115 --> 00:14:44,650

the printout which would just

398

00:14:44,650 --> 00:14:46,410

be 100 pages that said error,

399

00:14:46,410 --> 00:14:48,705

error, error and then

400

00:14:48,705 --> 00:14:51,590

you'd have to go back figure

out why it was that.

401

00:14:51,590 --> 00:14:54,180

It was horrible it was like

a week long debug cycle.

402

00:14:54,180 --> 00:14:54,455

>> Yeah.

403

00:14:54,455 --> 00:14:56,140

>> But then they

started getting some

404

00:14:56,140 --> 00:14:58,080

again also some Z-80 and

405

00:14:58,080 --> 00:15:01,410

CPM-based microcomputers and by

406

00:15:01,410 --> 00:15:03,855

the time I left

the university there,

407

00:15:03,855 --> 00:15:05,950

they were actually

using Turbo Pascal or

408

00:15:05,950 --> 00:15:07,430

Poly Pascal the one that I

409

00:15:07,430 --> 00:15:09,040

written in the

introductory courses.

410

00:15:09,040 --> 00:15:10,265

>> Holy crap. That must be great.

411

00:15:10,265 --> 00:15:12,650

>> So my professor

did allow me to not

412

00:15:12,650 --> 00:15:16,700

attend the class he gave me an A.

413

00:15:16,700 --> 00:15:18,760

>> Because you've

written the compiler.

414

00:15:18,760 --> 00:15:20,150

>> Yeah, that's right.

415

00:15:20,150 --> 00:15:21,515

>> That's great.

416

00:15:21,515 --> 00:15:22,360

>> Yeah.

417

00:15:22,360 --> 00:15:25,420

>> I think one of the things

that folks may have started

418

00:15:25,420 --> 00:15:28,620

to take for granted now is

like you sort of jump online,

419

00:15:28,620 --> 00:15:30,960

you've got Google, you've got

the open source community.

420

00:15:30,960 --> 00:15:31,700

>> Right.

421

00:15:31,700 --> 00:15:34,080

>> YouTube and like

all these video resources.

422

00:15:34,080 --> 00:15:35,860

I would argue that it's like much

423

00:15:35,860 --> 00:15:37,820

easier now to

bootstrap yourself as

424

00:15:37,820 --> 00:15:39,370

a programmer or even in

425

00:15:39,370 --> 00:15:42,025

sort of esoteric things like

how to build a compiler.

426

00:15:42,025 --> 00:15:43,640

So, how did you get

427

00:15:43,640 --> 00:15:44,850

your bootstrap, like how did

428

00:15:44,850 --> 00:15:46,420

you figure out how

to do this stuff?

429

00:15:46,420 --> 00:15:47,910

>> You learn by doing.

430

00:15:47,910 --> 00:15:49,770

You learn the hard way

in a sense.

431

00:15:49,770 --> 00:15:51,335

>> I mean in

retrospect there were

432

00:15:51,335 --> 00:15:53,320

so many things that

had I known that,

433

00:15:53,320 --> 00:15:55,195

I could have gotten

there a lot quicker.

434

00:15:55,195 --> 00:15:57,275

A couple of things I suppose is

435

00:15:57,275 --> 00:15:59,720

one is you couldn't

stop me from doing it.

436

00:15:59,720 --> 00:16:01,230

I wanted to do it. Someone once

437

00:16:01,230 --> 00:16:03,150

talked about how do you get

really good at something.

438

00:16:03,150 --> 00:16:04,820

Well, it's by putting

in the time. There's

439

00:16:04,820 --> 00:16:07,460

this talk about the

10,000 hours that you got

440

00:16:07,460 --> 00:16:10,965

to put in and I was like yeah

I definitely put in 10,000

441

00:16:10,965 --> 00:16:12,885

hours in the first three

years that I did

442

00:16:12,885 --> 00:16:15,065

this because I was obsessed.

443

00:16:15,065 --> 00:16:17,630

I just have to do it I didn't

care if anyone paid me.

444

00:16:17,630 --> 00:16:20,140

It was just a challenge

of it was so interesting.

445

00:16:20,140 --> 00:16:22,455

>> It probably wasn't

10,000 glorious hours.

446

00:16:22,455 --> 00:16:24,790

You were probably were

making lots of mistakes so.

447

00:16:24,790 --> 00:16:26,335

>> Absolutely. Absolutely. Yeah.

448

00:16:26,335 --> 00:16:27,690

I didn't know how to

write a compiler.

449

00:16:27,690 --> 00:16:28,980

But then you started looking at

450

00:16:28,980 --> 00:16:30,570

compilers that were

written and since

451

00:16:30,570 --> 00:16:31,900

my native language was

452

00:16:31,900 --> 00:16:34,410

machine code I mean heck

yeah I could disassemble

453

00:16:34,410 --> 00:16:36,210

other stuff and figure

out how that worked

454

00:16:36,210 --> 00:16:38,615

right and then you

gradually learned.

455

00:16:38,615 --> 00:16:40,030

But there were still

things I didn't know.

456

00:16:40,030 --> 00:16:41,325

Even in the first version of

457

00:16:41,325 --> 00:16:43,775

Turbo Pascal I remember

all the symbol tables were

458

00:16:43,775 --> 00:16:45,585

just stored as linear lists,

459

00:16:45,585 --> 00:16:47,855

which of course doesn't

scale so well, right?

460

00:16:47,855 --> 00:16:50,550

I mean and then I

remember learning

461

00:16:50,550 --> 00:16:53,380

about hash tables so I

was like, "Holy cow!

462

00:16:53,380 --> 00:16:54,505

I got to go home and try this."

463

00:16:54,505 --> 00:16:55,995

Then I put it in and doubled

464

00:16:55,995 --> 00:16:57,215

the speed of the compiler, right?

465

00:16:57,215 --> 00:17:00,025

I remember that's one

of the first books

466

00:17:00,025 --> 00:17:03,190

that I thought was

really instrumental for

467

00:17:03,190 --> 00:17:05,060

me in understanding

a lot of sort of

468

00:17:05,060 --> 00:17:07,105

the basic computer science

was written by

469

00:17:07,105 --> 00:17:09,304

Niklaus Wirth called Algorithms

470

00:17:09,304 --> 00:17:11,275

Plus Data Structures

Equals Programs.

471

00:17:11,275 --> 00:17:11,535

>> Yeah.

472

00:17:11,535 --> 00:17:12,150

>> Which is a great book.

473

00:17:12,150 --> 00:17:12,810

>> Oh my God.

474

00:17:12,810 --> 00:17:14,400

It is the most pragmatic

475

00:17:14,400 --> 00:17:16,665

computer science book

I have ever seen.

476

00:17:16,665 --> 00:17:18,050

I was like never into

477

00:17:18,050 --> 00:17:20,160

the symbolism and

the formal proofs and whatever.

478

00:17:20,160 --> 00:17:21,775

I'm just like give

me the algorithm.

479

00:17:21,775 --> 00:17:23,880

Tell me. Let me understand

how it works, right?

480

00:17:23,880 --> 00:17:24,045

>> Yeah.

481

00:17:24,045 --> 00:17:26,150

>> This thing explained how

482

00:17:26,150 --> 00:17:28,970

binary trees and

binary search and

483

00:17:28,970 --> 00:17:31,680

hash table and how

to build a compiler

484

00:17:31,680 --> 00:17:34,565

with a scanner and

a lexer and a parser.

485

00:17:34,565 --> 00:17:37,400

Do you know what I mean?

Code generator and

486

00:17:37,400 --> 00:17:41,335

recursion and oh my God it

was like I loved that book.

487

00:17:41,335 --> 00:17:41,960

>> Yeah.

488

00:17:41,960 --> 00:17:43,990

>> You can go back

to it today and it

489

00:17:43,990 --> 00:17:46,395

is still super valuable.

490

00:17:46,395 --> 00:17:48,140

>> I've always loved

books like that.

491

00:17:48,140 --> 00:17:50,680

Another good one Dave Hanson

492

00:17:50,680 --> 00:17:53,310

who used to work at

Microsoft Research here

493

00:17:53,310 --> 00:17:55,265

wrote this book called

C Interfaces and

494

00:17:55,265 --> 00:17:58,900

Implementations. And I like C

because it's sort of simple.

495

00:17:58,900 --> 00:18:00,500

I mean it's got a bunch

of hair on it, right?

496

00:18:00,500 --> 00:18:02,035

>> It does. It does.

497

00:18:02,035 --> 00:18:03,240

Easy-to-make mistakes. Yeah yeah.

498

00:18:03,240 --> 00:18:04,475

>> It's easy-to-make mistakes

499

00:18:04,475 --> 00:18:06,480

not the most elegant language but

500

00:18:06,480 --> 00:18:07,880

it's simple you can sort of pack

501

00:18:07,880 --> 00:18:10,125

all the rules and you add.

502

00:18:10,125 --> 00:18:13,650

But this book,

C Interfaces Implementations

503

00:18:13,650 --> 00:18:15,540

was genius because it

504

00:18:15,540 --> 00:18:18,140
was a bunch of data
structures and

505
00:18:18,140 --> 00:18:19,815
sort of approaches to programming

506
00:18:19,815 --> 00:18:21,020
that were super pragmatic.

507
00:18:21,020 --> 00:18:24,675
They had these very
broad applications

508
00:18:24,675 --> 00:18:25,860
and it was almost like

509
00:18:25,860 --> 00:18:27,490
teaching C to do

510
00:18:27,490 --> 00:18:30,975

crazy object-oriented sort

of things that couldn't-

511

00:18:30,975 --> 00:18:32,110

>> All your tables do.

512

00:18:32,110 --> 00:18:32,990

>> Yeah.

513

00:18:32,990 --> 00:18:34,605

>> Yeah, yeah no I

mean heck at Borland,

514

00:18:34,605 --> 00:18:38,675

we definitely had lots of

systems that were sort of calm

515

00:18:38,675 --> 00:18:41,370

that Windows is largely based

516

00:18:41,370 --> 00:18:44,365

on is effectively roll

your own read tables, right?

517

00:18:44,365 --> 00:18:47,805

>> At Borland, how did

you all get feedback?

518

00:18:47,805 --> 00:18:49,160

How did you know you were pushing

519

00:18:49,160 --> 00:18:51,985

your product in

the right direction?

520

00:18:51,985 --> 00:18:53,950

>> That's a good question.

521

00:18:53,950 --> 00:18:55,630

>> The cycles were long, right?

522

00:18:55,630 --> 00:18:57,200

>> They were. Back then there

523

00:18:57,200 --> 00:18:58,819

were a lot of trade magazines,

524

00:18:58,819 --> 00:19:01,055

and you got a lot of

feedback through that.

525

00:19:01,055 --> 00:19:02,720

Through the reviewers, like

526

00:19:02,720 --> 00:19:06,705

[inaudible] and Byte magazine

and what have you, right?

527

00:19:06,705 --> 00:19:10,115

We also had bulletin

boards early on.

528

00:19:10,115 --> 00:19:12,010

Like the old BBSes that we

529
00:19:12,010 --> 00:19:13,950
would monitor in
our tech support,

530
00:19:13,950 --> 00:19:16,490
and so, there were
a variety of ways.

531
00:19:16,490 --> 00:19:19,110
People would write us
letters and send us letters.

532
00:19:19,110 --> 00:19:22,315
I have so many letters with
suggestions for stuff.

533
00:19:22,315 --> 00:19:25,080
So, we always had a long list
of things we wanted to do.

534

00:19:25,080 --> 00:19:27,170

It was really not a matter

of figuring out what to do,

535

00:19:27,170 --> 00:19:29,275

it was like a matter of

which ones don't we do,

536

00:19:29,275 --> 00:19:31,930

and sorting it and

doing the right ones.

537

00:19:31,930 --> 00:19:34,225

You mentioned too Pascal 5.5.

538

00:19:34,225 --> 00:19:35,610

That was the release

where we introduced

539

00:19:35,610 --> 00:19:37,090

Object-Oriented Programming.

540

00:19:37,090 --> 00:19:39,580

Oh my God, that was like

the big buzz word of the time.

541

00:19:39,580 --> 00:19:42,895

That was like the AI

of the 80's, right?

542

00:19:42,895 --> 00:19:44,920

>> It was a good implementation.

543

00:19:44,920 --> 00:19:49,245

So, you must've taken

some inspiration from Modular.

544

00:19:49,245 --> 00:19:51,120

>> Well, Object Pascal was

545

00:19:51,120 --> 00:19:53,815

probably the work that

Larry Tesler did at Apple.

546

00:19:53,815 --> 00:19:55,795

That was definitely

an inspiration.

547

00:19:55,795 --> 00:19:58,980

Although, our first

implementation interwove Pascal

548

00:19:58,980 --> 00:20:00,770

5.5 was a little bit different.

549

00:20:00,770 --> 00:20:04,520

It had some influences

from C++ also.

550

00:20:04,520 --> 00:20:06,370

Then by the time

we got to Delphi,

551

00:20:06,370 --> 00:20:09,670
I think we were more
true to Object Pascal.

552
00:20:09,670 --> 00:20:12,450
There was no difference
between object references.

553
00:20:12,450 --> 00:20:14,790
You didn't have to use
an up arrow to dereference them,

554
00:20:14,790 --> 00:20:18,320
and then that made the syntax
a lot more gentle on user.

555
00:20:18,320 --> 00:20:21,845
It was just fu.bar
not fuuparrow.bar.

556
00:20:21,845 --> 00:20:24,050
>> Yeah, I remember.

So, 5.5 was where

557

00:20:24,050 --> 00:20:26,420

I learned Object-Oriented

Programming.

558

00:20:26,420 --> 00:20:29,490

I think I'm sort of lucky

because if you got thrown

559

00:20:29,490 --> 00:20:32,620

into a modern Java for instance,

560

00:20:32,620 --> 00:20:35,055

and that was your playground

for Object-Oriented.

561

00:20:35,055 --> 00:20:37,135

It's so complicated now.

562

00:20:37,135 --> 00:20:40,175

>> It is, but it's also in
some ways simpler, right?

563
00:20:40,175 --> 00:20:43,505
Because back then, the
languages that people used,

564
00:20:43,505 --> 00:20:45,965
by and large, did not have
garbage collection built-in,

565
00:20:45,965 --> 00:20:51,380
and were very not safe and had
minimal runtime checking.

566
00:20:51,380 --> 00:20:53,250
In fact, we ship Turbo Pascal by

567
00:20:53,250 --> 00:20:55,655
default with runtime
checking turned off,

568

00:20:55,655 --> 00:20:58,380

because we did better on

the benchmarks that way.

569

00:20:58,380 --> 00:20:59,800

A lot of reviewers were not

570

00:20:59,800 --> 00:21:01,440

smart enough to know

to turn it off,

571

00:21:01,440 --> 00:21:03,825

and so, they were comparing

apples to oranges.

572

00:21:03,825 --> 00:21:05,600

We just wanted to make sure that

573

00:21:05,600 --> 00:21:07,370

we were always in

the running there.

574

00:21:07,370 --> 00:21:09,630

Once you introduce

garbage collection,

575

00:21:09,630 --> 00:21:11,700

then you really start to

576

00:21:11,700 --> 00:21:14,095

up the level of

programming styles.

577

00:21:14,095 --> 00:21:15,950

You can write in

a style where you

578

00:21:15,950 --> 00:21:17,840

have much more complex

function results,

579

00:21:17,840 --> 00:21:20,990

and you don't have to worry

about who de-allocates this.

580

00:21:20,990 --> 00:21:22,150

Who owns this memory.

581

00:21:22,150 --> 00:21:23,805

Who's supposed to

get rid of it again,

582

00:21:23,805 --> 00:21:25,440

and you don't have all of

583

00:21:25,440 --> 00:21:27,760

these cycles that you

maybe get if you implement

584

00:21:27,760 --> 00:21:29,580

with Ref counting or whatever

585

00:21:29,580 --> 00:21:32,060

you do to keep track

of your objects.

586

00:21:32,060 --> 00:21:35,705

So, it made the world a lot

simpler in a lot of ways.

587

00:21:35,705 --> 00:21:39,100

Then, the fact that these

languages can be memory-safe,

588

00:21:39,100 --> 00:21:42,045

that removes

a whole bunch of other.

589

00:21:42,045 --> 00:21:44,880

The thing that saved

us all at the time

590

00:21:44,880 --> 00:21:47,390

was that there were

just only so much capacity.

591

00:21:47,390 --> 00:21:49,440

There's only so much

you can do in 64K.

592

00:21:49,440 --> 00:21:52,380

Even though Biulds and 64

was going to be enough,

593

00:21:52,380 --> 00:21:55,145

there's only so much

you can do on 640 also.

594

00:21:55,145 --> 00:21:56,770

So, that always saved us.

595

00:21:56,770 --> 00:21:59,340

It was like the

complexity was never big

596

00:21:59,340 --> 00:22:02,345

enough that you couldn't

get away with it.

597

00:22:02,345 --> 00:22:04,030

But once you're in this now,

598

00:22:04,030 --> 00:22:06,020

is literally a bottomless pit.

599

00:22:06,020 --> 00:22:08,570

There's so much code, and you

take so many dependencies,

600

00:22:08,570 --> 00:22:10,250

and if you don't have some of

601

00:22:10,250 --> 00:22:12,110

these guarantees built

into the system,

602

00:22:12,110 --> 00:22:13,460

you're going to be

spending an awful

603

00:22:13,460 --> 00:22:14,630

lot of time just chasing

604

00:22:14,630 --> 00:22:17,430

down dumb bugs that

really you shouldn't.

605

00:22:17,430 --> 00:22:19,240

That's not a creative process.

606

00:22:19,240 --> 00:22:20,600

That is just a waste of time.

607

00:22:20,600 --> 00:22:22,180

>> Yeah. So, I want

to get to that,

608

00:22:22,180 --> 00:22:25,035

but before we jump

into modern type,

609

00:22:25,035 --> 00:22:27,535

let's talk a little bit

about Delphi.

610

00:22:27,535 --> 00:22:29,750

So, maybe the compiler

didn't change

611

00:22:29,750 --> 00:22:32,010

all that much but

my god, that would be-

612

00:22:32,010 --> 00:22:34,640

>> That was a revolution

on the IDE side.

613

00:22:34,640 --> 00:22:36,290

It's interesting

because I've worked

614

00:22:36,290 --> 00:22:37,670

on a project in between.

615

00:22:37,670 --> 00:22:38,765

I never went anywhere.

616

00:22:38,765 --> 00:22:40,830

It was code-named

Monet at Borland,

617

00:22:40,830 --> 00:22:44,170

and it was about building

applications visually,

618

00:22:44,170 --> 00:22:46,240

which was very sort of- Back

619

00:22:46,240 --> 00:22:48,700

then everyone was talking

about software ICs,

620

00:22:48,700 --> 00:22:51,180

and you could just plug

and play and draw lines

621

00:22:51,180 --> 00:22:54,195

between the inputs and outputs

on the different things.

622

00:22:54,195 --> 00:22:56,495

We were building

this very advanced set of

623

00:22:56,495 --> 00:22:58,085

components that you

could just plunk

624

00:22:58,085 --> 00:23:00,000

down on a canvas

and wire together,

625

00:23:00,000 --> 00:23:01,460
and making Fahrenheit to

626
00:23:01,460 --> 00:23:03,220
centigrade converters
was super easy.

627
00:23:03,220 --> 00:23:05,940
But it turned out that
making big apps was

628
00:23:05,940 --> 00:23:07,090
horrible because you had

629
00:23:07,090 --> 00:23:08,680
lines going from
everything to everything,

630
00:23:08,680 --> 00:23:10,325
and it just didn't scale.

631

00:23:10,325 --> 00:23:12,815

But I think in that process,

632

00:23:12,815 --> 00:23:14,600

I got a real appreciation

633

00:23:14,600 --> 00:23:17,650

for what does work well visually.

634

00:23:17,650 --> 00:23:21,235

Like designing input

and GUIs and whatever,

635

00:23:21,235 --> 00:23:23,080

and GUI was starting to happen.

636

00:23:23,080 --> 00:23:25,435

So, there was definitely

a need there.

637

00:23:25,435 --> 00:23:28,675

So, by the time this Monet
thing did not pan out,

638
00:23:28,675 --> 00:23:30,280
we realized it wasn't
going to pan out,

639
00:23:30,280 --> 00:23:32,700
then we went back to basics.

640
00:23:32,700 --> 00:23:34,690
Then go, "Okay. Well,
let's actually get

641
00:23:34,690 --> 00:23:36,900
into the GUI era
with this product."

642
00:23:36,900 --> 00:23:39,180
Then there was this thing
called Visual Basic out.

643

00:23:39,180 --> 00:23:39,705

>> Correct.

644

00:23:39,705 --> 00:23:42,730

>> That taught us a lot, but

also had a lot of issues.

645

00:23:42,730 --> 00:23:45,224

Like it didn't have

object orientation,

646

00:23:45,224 --> 00:23:47,000

it was interpreter-based,

647

00:23:47,000 --> 00:23:48,850

it was generally frowned

648

00:23:48,850 --> 00:23:50,860

upon language at least

by a lot of people,

649

00:23:50,860 --> 00:23:52,880

and so, we go, "We can do this.

650

00:23:52,880 --> 00:23:54,655

We can do this, and

we can do it better."

651

00:23:54,655 --> 00:23:55,210

>> Yeah.

652

00:23:55,210 --> 00:23:57,295

>> That was the

genesis for Delphi.

653

00:23:57,295 --> 00:24:00,835

Then I think in the process

of building Delphi,

654

00:24:00,835 --> 00:24:02,940

we also realized that-

655

00:24:02,940 --> 00:24:05,355

and there is this I

think was Zach Urbach,

656

00:24:05,355 --> 00:24:08,290

our program manager who astutely

657

00:24:08,290 --> 00:24:11,300

realized that you can't

just ship a Rad tool.

658

00:24:11,300 --> 00:24:12,710

You've got to have some angle,

659

00:24:12,710 --> 00:24:15,050

and the angle at the

time was client-server.

660

00:24:15,050 --> 00:24:19,410

So, we actually pivoted it

into a client-server tool.

661

00:24:19,410 --> 00:24:23,230

That gave us an inroad

into enterprises and

662

00:24:23,230 --> 00:24:25,200

companies that were

really building

663

00:24:25,200 --> 00:24:27,410

real database connected apps.

664

00:24:27,410 --> 00:24:29,580

I remember competing

with PowerBuilder.

665

00:24:29,580 --> 00:24:31,320

I don't know if you

were with PowerBuilder.

666

00:24:31,320 --> 00:24:34,325

So, that gave us
a unique vantage point

667
00:24:34,325 --> 00:24:35,920
compared to Visual Basic.

668
00:24:35,920 --> 00:24:38,085
>> It was sort of an
interesting thing.

669
00:24:38,085 --> 00:24:42,950
Delphi came out shortly after
I graduated from college,

670
00:24:42,950 --> 00:24:45,500
and I became a real
software engineer,

671
00:24:45,500 --> 00:24:47,940
I remember being
so grateful for it

672

00:24:47,940 --> 00:24:50,650

because I did a bunch

of work with it

673

00:24:50,650 --> 00:24:52,625

that clients were paying me for,

674

00:24:52,625 --> 00:24:56,285

and the tool made me look

better than I actually was.

675

00:24:56,285 --> 00:24:59,030

We had made this transition from,

676

00:24:59,030 --> 00:25:00,850

you build in

these text-based apps

677

00:25:00,850 --> 00:25:02,170

where if you really fancy,

678

00:25:02,170 --> 00:25:03,355

you're sort of blasted

679

00:25:03,355 --> 00:25:05,665

ASCII codes and

the screen buffers.

680

00:25:05,665 --> 00:25:07,955

But Windows 95 came out.

681

00:25:07,955 --> 00:25:09,760

The onus was on you.

It looks slick.

682

00:25:09,760 --> 00:25:11,550

It was a pretty Operating System.

683

00:25:11,550 --> 00:25:13,755

>> It was an awesome way

to build client apps.

684

00:25:13,755 --> 00:25:14,910

It really was, and then it had

685

00:25:14,910 --> 00:25:16,320

a bunch of fun things in it.

686

00:25:16,320 --> 00:25:18,565

I remember the two-way

code generator.

687

00:25:18,565 --> 00:25:21,020

You can either visually design

your form or you could go

688

00:25:21,020 --> 00:25:22,130

in and modify the code and

689

00:25:22,130 --> 00:25:24,450

then the form would

change, right?

690

00:25:24,590 --> 00:25:28,500

Making that work was not easy,

691

00:25:28,500 --> 00:25:29,615

but once it did work,

692

00:25:29,615 --> 00:25:31,120

oh my God, it was so cool.

693

00:25:31,120 --> 00:25:33,240

>> That must be

a gratifying thing.

694

00:25:33,240 --> 00:25:35,390

You've been doing this

your entire career,

695

00:25:35,390 --> 00:25:38,470

but in my mind,

696

00:25:38,470 --> 00:25:40,560
a force multiplying
effect of building

697
00:25:40,560 --> 00:25:42,300
developer tools as you put

698
00:25:42,300 --> 00:25:44,060
your development effort
into this thing,

699
00:25:44,060 --> 00:25:45,120
and then it gets into

700
00:25:45,120 --> 00:25:47,160
the hands of the huge
number of people.

701
00:25:47,160 --> 00:25:48,480
>> That is the thing that I

702

00:25:48,480 --> 00:25:52,020
think in retrospect has
been hugely rewarding.

703
00:25:52,020 --> 00:25:54,355
It's like when you
build end user apps,

704
00:25:54,355 --> 00:25:57,140
you don't build the same kind
of fandom as you

705
00:25:57,140 --> 00:25:58,360
do when you're building

706
00:25:58,360 --> 00:26:00,710
developer tools.
Because developers,

707
00:26:00,710 --> 00:26:02,825
they put in the hours.
Do you know what I mean?

708

00:26:02,825 --> 00:26:02,990

>> Yeah.

709

00:26:02,990 --> 00:26:05,440

>> They come to really

depend on their tool.

710

00:26:05,440 --> 00:26:09,060

They come to truly

appreciate what they do.

711

00:26:09,060 --> 00:26:09,560

>> Yes.

712

00:26:09,560 --> 00:26:12,170

>> Then they become

these ardent fans.

713

00:26:12,170 --> 00:26:13,590

>> They are almost

religious, right?

714

00:26:13,590 --> 00:26:15,355

>> Yeah, I know, and

that is so rewarding.

715

00:26:15,355 --> 00:26:17,040

Throughout my career,

that has been so

716

00:26:17,040 --> 00:26:18,800

rewarding to go speak

at a conference and

717

00:26:18,800 --> 00:26:20,260

have all these people

come up and tell me

718

00:26:20,260 --> 00:26:23,075

how you saved my life.

You did this.

719

00:26:23,075 --> 00:26:24,890

People have literally

told men that.

720

00:26:24,890 --> 00:26:27,260

It's like guys who

were in Russia,

721

00:26:27,260 --> 00:26:28,790

and couldn't put

bread on the table

722

00:26:28,790 --> 00:26:30,710

learned to program with

Turbo Pascal and

723

00:26:30,710 --> 00:26:34,370

literally this guy is claiming

I saved his family's life.

724

00:26:34,370 --> 00:26:35,865

I mean, that I'm just God.

725

00:26:35,865 --> 00:26:37,845

That's amazing.

That is wonderful.

726

00:26:37,845 --> 00:26:38,745

>> Really great.

727

00:26:38,745 --> 00:26:39,125

>> Yeah.

728

00:26:39,125 --> 00:26:42,500

>> So, at the time you sort

of referenced it already,

729

00:26:42,500 --> 00:26:45,025

there was this

intense competition

730

00:26:45,025 --> 00:26:46,900

amongst commercial entities,

731
00:26:46,900 --> 00:26:49,130
companies like Borland
and Microsoft,

732
00:26:49,130 --> 00:26:51,555
and the guys who were
building PowerBuilder.

733
00:26:51,555 --> 00:26:53,720
There were these
database language things

734
00:26:53,720 --> 00:26:55,730
going around and literally

735
00:26:55,730 --> 00:26:58,490
catalogs full of
shrink-wrapped software

736

00:26:58,490 --> 00:26:59,540
that was just for

737
00:26:59,540 --> 00:27:01,595
developers to help them
get their job done.

738
00:27:01,595 --> 00:27:04,090
So, the big thing that came on

739
00:27:04,090 --> 00:27:06,715
the scene right after
Delphi was Java.

740
00:27:06,715 --> 00:27:09,250
How did that factor
into your move

741
00:27:09,250 --> 00:27:12,325
from Borland to Microsoft?

742

00:27:12,325 --> 00:27:14,620

Because, in a way C-Sharp,

it seems like [inaudible].

743

00:27:14,620 --> 00:27:17,285

>> There is an arch

there definitely.

744

00:27:17,285 --> 00:27:18,320

But I do remember,

745

00:27:18,320 --> 00:27:19,885

I think it was in '95 when

746

00:27:19,885 --> 00:27:22,825

Java made its first appearance.

747

00:27:22,825 --> 00:27:25,450

This is where the Internet

was starting to happen.

748

00:27:25,450 --> 00:27:27,270
Particularly, I think in

749
00:27:27,270 --> 00:27:29,170
the development tool space
and programming,

750
00:27:29,170 --> 00:27:32,450
there was this collective
madness around Java

751
00:27:32,450 --> 00:27:35,905
where everyone thought it
was over. It was done.

752
00:27:35,905 --> 00:27:37,580
There were going to
be no more [inaudible] and

753
00:27:37,580 --> 00:27:39,590
Java to take over
the whole world.

754

00:27:39,590 --> 00:27:41,365

There was a Java fund.

755

00:27:41,365 --> 00:27:42,960

Remember the Java fund?

756

00:27:42,960 --> 00:27:45,190

Funded by hundreds of

millions of dollars

757

00:27:45,190 --> 00:27:47,770

whose sole purpose

was to just invest

758

00:27:47,770 --> 00:27:50,775

in companies that were building

their software in Java.

759

00:27:50,775 --> 00:27:52,560

As if that would

make a difference.

760

00:27:52,560 --> 00:27:54,385

It was brilliant, what Sun did,

761

00:27:54,385 --> 00:27:56,170

I think that even

they had no idea

762

00:27:56,170 --> 00:27:57,980

what they had

unleashed here, right?

763

00:27:57,980 --> 00:27:59,480

But it was discouraging in

764

00:27:59,480 --> 00:28:02,340

many ways because we felt that,

765

00:28:02,340 --> 00:28:04,370

yes, that's fantastic for

766

00:28:04,370 --> 00:28:05,860

building applets in the browser.

767

00:28:05,860 --> 00:28:08,390

But you can't build

real Apps with this thing.

768

00:28:08,390 --> 00:28:09,610

But hey, you know what?

769

00:28:09,610 --> 00:28:11,170

There were the shiny object.

770

00:28:11,170 --> 00:28:14,760

There was collective craziness

over the Java thing,

771

00:28:14,760 --> 00:28:15,890

but there were also things

772

00:28:15,890 --> 00:28:17,425

that I thought were interesting.

773

00:28:17,425 --> 00:28:19,600

This whole notion

of write once and

774

00:28:19,600 --> 00:28:21,900

run everywhere is even today,

775

00:28:21,900 --> 00:28:23,260

it just so happened that it

776

00:28:23,260 --> 00:28:24,710

wasn't actually born out by Java.

777

00:28:24,710 --> 00:28:26,335

It was born out by Javascript.

778

00:28:26,335 --> 00:28:28,660

But, which was there already

779

00:28:28,660 --> 00:28:31,140

in the browser and

ignored for a decade.

780

00:28:31,140 --> 00:28:34,740

It's crazy. We'd had a JBuilder

tool we built at Borland,

781

00:28:34,740 --> 00:28:36,975

which actually was

built in Delphi.

782

00:28:36,975 --> 00:28:39,325

They started with the Delphi IDE,

783

00:28:39,325 --> 00:28:42,550

chopped it down, and then the

language service, well-

784

00:28:42,550 --> 00:28:45,065
They weren't really called
Language Services at the time,

785
00:28:45,065 --> 00:28:48,690
but the Mini compiler analyzer
are used by the IDE to

786
00:28:48,690 --> 00:28:51,110
do all of their
syntactic highlighting

787
00:28:51,110 --> 00:28:53,450
and whatever that was
also written in Delphi.

788
00:28:53,450 --> 00:28:57,185
So, I got to know
Java as a language.

789
00:28:57,185 --> 00:28:58,700
Then at the same time,

790

00:28:58,700 --> 00:29:00,560

there was a change afoot at Borland.

791

00:29:00,560 --> 00:29:02,840

Borland had a bit of

an identity crisis,

792

00:29:02,840 --> 00:29:04,880

in the sense that everyone

793

00:29:04,880 --> 00:29:07,370

was up leveling how

business was done.

794

00:29:07,370 --> 00:29:09,990

It all became a game

of platforms.

795

00:29:09,990 --> 00:29:11,830

Oracle was databases.

796

00:29:11,830 --> 00:29:13,920

Sun was workstations.

797

00:29:13,920 --> 00:29:16,985

Microsoft was the OS,

798

00:29:16,985 --> 00:29:19,870

and then Office became

another platform.

799

00:29:19,870 --> 00:29:24,420

And Borland never really got

the platform up leveling.

800

00:29:24,420 --> 00:29:26,750

It was always a bunch

of separate products,

801

00:29:26,750 --> 00:29:28,445

and we try to bundle

them together,

802

00:29:28,445 --> 00:29:30,605

but it never really made sense.

803

00:29:30,605 --> 00:29:34,410

Then the company was so

singularly focused on trying to

804

00:29:34,410 --> 00:29:36,840

get a foothold in

that business that

805

00:29:36,840 --> 00:29:39,110

they started neglecting

their developer tools,

806

00:29:39,110 --> 00:29:41,090

which was really where

the company came from,

807

00:29:41,090 --> 00:29:43,825
and a series of
management revolving door.

808
00:29:43,825 --> 00:29:45,565
Then I got more and
more disillusioned.

809
00:29:45,565 --> 00:29:47,980
Then of course,
Borland ended up going

810
00:29:47,980 --> 00:29:51,090
through a bunch of reductions
in force if you will,

811
00:29:51,090 --> 00:29:54,210
and we spread a bunch of
people into the universe.

812
00:29:54,210 --> 00:29:55,980
A bunch of them

went to the valley.

813

00:29:55,980 --> 00:29:57,635

A lot of them went to Microsoft.

814

00:29:57,635 --> 00:29:59,795

Then of course, they

started calling,

815

00:29:59,795 --> 00:30:02,000

every summer when the weather

is nice in Seattle,

816

00:30:02,000 --> 00:30:04,165

I'd get a call from

Brad Silverberg,

817

00:30:04,165 --> 00:30:06,590

who had run R&D at Borland,

818

00:30:06,590 --> 00:30:09,570

and now was running

Windows 95 development.

819

00:30:09,570 --> 00:30:11,975

So, I would come up and we'd talk,

820

00:30:11,975 --> 00:30:13,660

and the first year I

wasn't interested.

821

00:30:13,660 --> 00:30:16,395

But then eventually, it was time.

822

00:30:16,395 --> 00:30:19,070

So, I joined Microsoft in '96.

823

00:30:19,070 --> 00:30:20,900

>> Was C# the first thing

that you worked on?

824

00:30:20,900 --> 00:30:22,840

>> No. Not at all.

There was no C#

825

00:30:22,840 --> 00:30:24,325

There were.Net at that time.

826

00:30:24,325 --> 00:30:25,840

I joined to work on

827

00:30:25,840 --> 00:30:27,840

Microsoft's Java

development tool set.

828

00:30:27,840 --> 00:30:28,470

>> Interesting.

829

00:30:28,470 --> 00:30:31,820

>> Yes, and I've

worked on Visual J++.

830

00:30:31,820 --> 00:30:33,050

So, at the time,

831

00:30:33,050 --> 00:30:35,450

we had a Java development tool,

832

00:30:35,450 --> 00:30:38,310

and it was really

just C++ IDE with

833

00:30:38,310 --> 00:30:39,990

the C++ compiler taken out and

834

00:30:39,990 --> 00:30:41,745

a Java compiler stuck

in there, right?

835

00:30:41,745 --> 00:30:42,090

>> Yes.

836

00:30:42,090 --> 00:30:45,375

>> That was visual J++

1.1 I think.

837

00:30:45,375 --> 00:30:47,790

Then, we jumped from 1.1 to

838

00:30:47,790 --> 00:30:50,970

6.0 because we realigned

all of our Rad tools,

839

00:30:50,970 --> 00:30:53,140

VB was at version 6, and so,

840

00:30:53,140 --> 00:30:55,810

it became Visual J++ 6.0,

841

00:30:55,810 --> 00:30:58,155

and it had a Visual designer.

842

00:30:58,155 --> 00:31:00,100

You could design

applets and whatever.

843

00:31:00,100 --> 00:31:02,645

But then we were also

doing stuff that made it

844

00:31:02,645 --> 00:31:05,200

a better language for

writing Windows Apps,

845

00:31:05,200 --> 00:31:07,675

and that ended up being

very controversial.

846

00:31:07,675 --> 00:31:10,390

I have personally read

the contract we had with Sun.

847

00:31:10,390 --> 00:31:12,675

It was explicitly

permitted in there that

848

00:31:12,675 --> 00:31:15,270
as long as we ran
all the standard tests,

849
00:31:15,270 --> 00:31:18,685
we were free to do
additional inventions.

850
00:31:18,685 --> 00:31:20,915
But that blew up.

851
00:31:20,915 --> 00:31:22,650
Literally, within three months of

852
00:31:22,650 --> 00:31:25,090
our shipping Visual J++,

853
00:31:25,090 --> 00:31:28,600
our product was enjoined
by a judge in San Jose,

854

00:31:28,600 --> 00:31:30,785
and we were required to put in

855
00:31:30,785 --> 00:31:33,430
a warning dialog, "Warning!

856
00:31:33,430 --> 00:31:34,650
You are about to turn

857
00:31:34,650 --> 00:31:37,895
on Microsoft
proprietary extensions.

858
00:31:37,895 --> 00:31:41,070
Are you certain
your wish to proceed?"

859
00:31:41,460 --> 00:31:45,315
And that was just so you
could use com interop.

860

00:31:45,315 --> 00:31:47,330

That was considered so evil.

861

00:31:47,330 --> 00:31:48,835

I was like this is crazy.

862

00:31:48,835 --> 00:31:51,735

Inside Microsoft at

the time, the Java fever,

863

00:31:51,735 --> 00:31:53,390

it was all over the industry and

864

00:31:53,390 --> 00:31:55,020

there were literally fractions of

865

00:31:55,020 --> 00:31:56,300

groups at Microsoft that

866

00:31:56,300 --> 00:31:58,420

believed that it was

all going to be Java.

867

00:31:58,420 --> 00:31:59,970

We should build a Java OS,

868

00:31:59,970 --> 00:32:02,090

it should all just be Java.

869

00:32:02,090 --> 00:32:05,500

But then that happened the

whole Sun lawsuit and whatever

870

00:32:05,500 --> 00:32:07,030

and that was sort of the genesis

871

00:32:07,030 --> 00:32:08,600

for us understanding that,

872

00:32:08,600 --> 00:32:11,690

listen, it's really hard to do

873

00:32:11,690 --> 00:32:13,350

what's right for

your customers by

874

00:32:13,350 --> 00:32:15,500

extending someone

else's platform.

875

00:32:15,500 --> 00:32:17,960

We've got to build our

own that works for

876

00:32:17,960 --> 00:32:19,450

what it is that our customers

877

00:32:19,450 --> 00:32:20,690

are telling us that they want.

878

00:32:20,690 --> 00:32:22,950

>> That was the genesis

of dot net and of course,

879

00:32:22,950 --> 00:32:24,470

we needed a programming language.

880

00:32:24,470 --> 00:32:26,585

I so happened to find myself

881

00:32:26,585 --> 00:32:29,485

at the right time at

the right place for that.

882

00:32:29,485 --> 00:32:31,190

>> So how fun was that?

883

00:32:31,190 --> 00:32:34,295

Is that the first time that

you had a blank slate?

884

00:32:34,295 --> 00:32:36,180

>> Totally. Yeah. I

mean can you imagine

885

00:32:36,180 --> 00:32:38,400

the opportunity to have a company

886

00:32:38,400 --> 00:32:40,225

like Microsoft put their might

887

00:32:40,225 --> 00:32:41,860

behind a programming language

888

00:32:41,860 --> 00:32:42,665

that you're going to create.

889

00:32:42,665 --> 00:32:45,620

I was flabbergasted that I

was given that opportunity.

890

00:32:45,620 --> 00:32:47,740

Yeah, sure I'd proven

myself but there

891

00:32:47,740 --> 00:32:49,730

were still a lot of

other people at Microsoft

892

00:32:49,730 --> 00:32:51,935

who had different ideas

about what should be done

893

00:32:51,935 --> 00:32:53,210

and then ultimately I

894

00:32:53,210 --> 00:32:54,730

was the one who was

handed the task.

895

00:32:54,730 --> 00:32:56,360

So that was wonderful and we had

896

00:32:56,360 --> 00:32:58,460

a great team on that early

897

00:32:58,460 --> 00:33:00,210

C sharp compiler which originally

898

00:33:00,210 --> 00:33:02,140

was written in C++ or C plus

899

00:33:02,140 --> 00:33:06,310

minus because we didn't

use all the C++.

900

00:33:06,310 --> 00:33:07,945

>> Which is a whole other design.

901

00:33:07,945 --> 00:33:10,010

>> Yeah, yeah, I

remember a guy named

902

00:33:10,010 --> 00:33:11,085

Peter Goldie wrote

903

00:33:11,085 --> 00:33:13,090
the core implementation
of the compiler,

904
00:33:13,090 --> 00:33:14,240
I wrote the language spec.

905
00:33:14,240 --> 00:33:17,070
We had a wonderful design
team process that we set

906
00:33:17,070 --> 00:33:20,755
up that you know to
this day still exists.

907
00:33:20,755 --> 00:33:22,455
Where we used to have

908
00:33:22,455 --> 00:33:25,035
three meetings a week
for two hours,

909

00:33:25,035 --> 00:33:26,915

where we would just do

the language design.

910

00:33:26,915 --> 00:33:28,565

Iterate on it, you know what I mean?

911

00:33:28,565 --> 00:33:31,590

And Scott Wilson with

our program manager

912

00:33:31,590 --> 00:33:34,895

that I worked with for

almost a decade was the PM.

913

00:33:34,895 --> 00:33:37,360

So for four years he

commuted from Hawaii.

914

00:33:37,360 --> 00:33:39,180

We would do

our design meetings on

915

00:33:39,180 --> 00:33:41,750

speakerphone with him.

It was a great time.

916

00:33:41,750 --> 00:33:44,260

>> Based on some of the stuff

that you've talked about

917

00:33:44,260 --> 00:33:45,590

already and just sort of

918

00:33:45,590 --> 00:33:47,635

looking at C sharp

and then typescript,

919

00:33:47,635 --> 00:33:50,940

you're fairly pragmatic when

it comes to language design.

920

00:33:50,940 --> 00:33:52,780
You're not one of these language
designers is like, "Oh,

921
00:33:52,780 --> 00:33:54,840
let's throw a Haylie Milnor
type system and-"

922
00:33:54,840 --> 00:33:58,800
>> I've always been
a believer in understanding

923
00:33:58,800 --> 00:34:01,800
the problem and
feeling the utility

924
00:34:01,800 --> 00:34:04,890
of something before I
want to go implement it.

925
00:34:04,890 --> 00:34:06,770
I have to believe in it.

926

00:34:06,770 --> 00:34:08,390

I don't just throw

it in there, yeah,

927

00:34:08,390 --> 00:34:11,105

I need a type system, want

an Haylie Milnor or whatever.

928

00:34:11,105 --> 00:34:13,160

Now, no. I want to

understand why is it

929

00:34:13,160 --> 00:34:15,590

there and does it really

need to be there?

930

00:34:15,590 --> 00:34:17,750

>> And a real programmer is

going to benefit from it.

931

00:34:17,750 --> 00:34:19,680

>> Exactly and so that

was the case in C

932

00:34:19,680 --> 00:34:22,195

sharp and that drove

a lot of our decisions.

933

00:34:22,195 --> 00:34:24,800

For example, we

added properties as

934

00:34:24,800 --> 00:34:26,505

a first class language construct

935

00:34:26,505 --> 00:34:28,820

because everyone was

doing properties.

936

00:34:28,820 --> 00:34:30,750

It was like that was

how programs were

937

00:34:30,750 --> 00:34:33,825

written and Java didn't

have those, for example.

938

00:34:33,825 --> 00:34:36,700

So there were pragmatic

decisions that we made.

939

00:34:36,700 --> 00:34:38,430

So whenever someone goes,

940

00:34:38,430 --> 00:34:39,590

why is this feature there?

941

00:34:39,590 --> 00:34:41,435

It's not just because

I thought it was fun,

942

00:34:41,435 --> 00:34:44,640

it's because it solves this,

this and this problem.

943

00:34:44,640 --> 00:34:47,225

And ideally, with every

language feature,

944

00:34:47,225 --> 00:34:49,330

this is something you

come to learn over time

945

00:34:49,330 --> 00:34:51,910

is, people will request a lot of

946

00:34:51,910 --> 00:34:53,870

features but they

will always have

947

00:34:53,870 --> 00:34:57,505

synthesized their idea of

what it is that they want.

948

00:34:57,505 --> 00:35:01,220
They really come to you
with an instance of a class

949
00:35:01,220 --> 00:35:03,070
of problem and they tell you I

950
00:35:03,070 --> 00:35:05,000
need this instance. And then-.

951
00:35:05,000 --> 00:35:06,250
>> So the trick is to get them to

952
00:35:06,250 --> 00:35:07,510
really describe the problem.

953
00:35:07,510 --> 00:35:09,630
>> No, the trick is for you to

954
00:35:09,630 --> 00:35:11,870
understand what is

the class of problem.

955

00:35:11,870 --> 00:35:12,090

>> Got you.

956

00:35:12,090 --> 00:35:13,830

>> And that's what

you go implement.

957

00:35:13,830 --> 00:35:16,230

So I never wanted to

implement a language feature

958

00:35:16,230 --> 00:35:19,090

unless I could see

multiple good uses for it.

959

00:35:19,090 --> 00:35:21,365

We're not just going to

implement because it does

960

00:35:21,365 --> 00:35:25,080
this one little corner of
XML well and that's it.

961
00:35:25,080 --> 00:35:28,290
I want this to be
a generally useful thing

962
00:35:28,290 --> 00:35:30,840
that you can do
other things with as well.

963
00:35:30,840 --> 00:35:32,720
>> Because it's
a very serious thing

964
00:35:32,720 --> 00:35:34,520
to bake something
into a language.

965
00:35:34,520 --> 00:35:37,020
>> The thing you come to learn

about languages is you can

966

00:35:37,020 --> 00:35:39,725

add all you want but you can

never take anything away.

967

00:35:39,725 --> 00:35:42,295

At least if you care about

backwards compatibility.

968

00:35:42,295 --> 00:35:44,190

And we've seen lots of

969

00:35:44,190 --> 00:35:46,460

train wrecks where

people did not care.

970

00:35:46,460 --> 00:35:48,070

Like say the transition from

971

00:35:48,070 --> 00:35:50,420

Python two to three

even when you're

972

00:35:50,420 --> 00:35:52,730

just doing it in the name

of cleaning up stuff

973

00:35:52,730 --> 00:35:55,630

that isn't right or

isn't ideal or whatever.

974

00:35:55,630 --> 00:35:58,080

You can't do that

to your user base.

975

00:35:58,080 --> 00:35:59,900

They have too much

of an investment,

976

00:35:59,900 --> 00:36:03,300

it is your responsibility

to ensure that their code

977

00:36:03,300 --> 00:36:05,010

continues to compile and

978

00:36:05,010 --> 00:36:06,780

work and then you

can add more on top.

979

00:36:06,780 --> 00:36:08,165

And if you're not

going to do that

980

00:36:08,165 --> 00:36:10,000

then you might as well

blow it all up and

981

00:36:10,000 --> 00:36:11,630

go create a whole new thing

982

00:36:11,630 --> 00:36:13,260

that advances

the state of the art.

983

00:36:13,260 --> 00:36:17,510

The worst thing you can

do is the 99% compatible,

984

00:36:17,510 --> 00:36:20,000

that is absolutely

the worst thing.

985

00:36:20,000 --> 00:36:21,930

>> So some of

that attachment to the

986

00:36:21,930 --> 00:36:24,180

old is emotional but you can very

987

00:36:24,180 --> 00:36:26,740

quickly have hundreds of millions

988

00:36:26,740 --> 00:36:28,870

of lines of code across.

989

00:36:28,870 --> 00:36:30,810

>> If you're a serving

enterprise, well,

990

00:36:30,810 --> 00:36:33,890

it doesn't even matter

enterprise is like anyone who

991

00:36:33,890 --> 00:36:35,370

writes a large body of code

992

00:36:35,370 --> 00:36:37,415

and then gets adopted somewhere,

993

00:36:37,415 --> 00:36:39,995

does not want to have to

spend a bunch of time

994

00:36:39,995 --> 00:36:42,990

changing it in random ways

just so it's prettier.

995

00:36:42,990 --> 00:36:45,925

No, big code bases get

ugly, they all do.

996

00:36:45,925 --> 00:36:48,210

Then programming

languages do too.

997

00:36:48,210 --> 00:36:50,450

I sometimes joke that show me

998

00:36:50,450 --> 00:36:51,980

the perfect programming

language and I'll

999

00:36:51,980 --> 00:36:53,690

show you a language

with no users,

1000

00:36:53,690 --> 00:36:55,880

because only then can

you go change it.

1001

00:36:55,880 --> 00:36:58,430

>> I did an internship

at Cray Research when I

1002

00:36:58,430 --> 00:37:01,160

was a younger engineer

still in grad school.

1003

00:37:01,160 --> 00:37:03,880

One of the most mind blowing

things that I had ever

1004

00:37:03,880 --> 00:37:05,310

heard up to that point

1005

00:37:05,310 --> 00:37:07,370

was I was sitting

at lunch one day,

1006
00:37:07,370 --> 00:37:09,610
one of these Cray
veteran engineers

1007
00:37:09,610 --> 00:37:11,740
was describing
this sales process.

1008
00:37:11,740 --> 00:37:12,960
So, they were trying to sell

1009
00:37:12,960 --> 00:37:14,880
a supercomputer at some point to

1010
00:37:14,880 --> 00:37:16,790
a car manufacturer and

1011
00:37:16,790 --> 00:37:19,270
because they had

this big 4 train codebase,

1012

00:37:19,270 --> 00:37:21,120

it was due in

finite element analysis.

1013

00:37:21,120 --> 00:37:24,290

So they could simulate crashes

on these supercomputers.

1014

00:37:24,290 --> 00:37:28,099

They'd come out with this

fancy new version of unicode,

1015

00:37:28,099 --> 00:37:29,810

so it was just going

to be impossible

1016

00:37:29,810 --> 00:37:31,350

for the car company to

1017

00:37:31,350 --> 00:37:34,740
rewrite their 4 train
code base which was tens and

1018
00:37:34,740 --> 00:37:38,150
tens of millions of
lines of code to work on

1019
00:37:38,150 --> 00:37:39,420
this operating system for

1020
00:37:39,420 --> 00:37:41,830
this new machine and the
guy's looking at it like,

1021
00:37:41,830 --> 00:37:43,540
all right, this machine
is millions of dollars

1022
00:37:43,540 --> 00:37:45,490
and that's tens of millions
of lines of code.

1023

00:37:45,490 --> 00:37:48,995

The operating system kernel

is 25,000 lines of code,

1024

00:37:48,995 --> 00:37:51,800

we'll just write a custom

operating system for this.

1025

00:37:51,800 --> 00:37:53,420

>> Being able to unpack where

1026

00:37:53,420 --> 00:37:55,570

the abstraction

and complexity is,

1027

00:37:55,570 --> 00:37:57,880

is really sort of an

interesting skill

1028

00:37:57,880 --> 00:37:59,290

you have to develop

as an engineer.

1029

00:37:59,290 --> 00:38:01,990

>> Yeah. So pragmatism was always

1030

00:38:01,990 --> 00:38:05,250

a main driver for me

and it continues to be.

1031

00:38:05,250 --> 00:38:07,695

>> What lessons is

a language designer

1032

00:38:07,695 --> 00:38:10,190

or like as a builder

of development tools,

1033

00:38:10,190 --> 00:38:11,730

have you learned that you're

1034

00:38:11,730 --> 00:38:13,770
applying the
typescript right now?

1035
00:38:13,770 --> 00:38:15,995
>> Well, we touched
on a couple of them.

1036
00:38:15,995 --> 00:38:17,845
You really have to respect

1037
00:38:17,845 --> 00:38:20,110
and value backwards
compatibility.

1038
00:38:20,110 --> 00:38:22,030
Doing language design,
it's interesting,

1039
00:38:22,030 --> 00:38:23,310
there's really in a sense

1040

00:38:23,310 --> 00:38:25,140

two phases to

designing a language.

1041

00:38:25,140 --> 00:38:27,970

There's 1.0 and then

there's everything else.

1042

00:38:27,970 --> 00:38:29,865

And with 1.0

it's a greenfield.

1043

00:38:29,865 --> 00:38:31,565

You can do whatever

you want to do.

1044

00:38:31,565 --> 00:38:34,320

It's crazy fun and you

can be super inventive.

1045

00:38:34,320 --> 00:38:36,400

But then from there

on out it becomes

1046

00:38:36,400 --> 00:38:38,420

a very different

discipline because now,

1047

00:38:38,420 --> 00:38:41,640

it's a game of trying to

sneak in new features

1048

00:38:41,640 --> 00:38:45,130

in a way that does not break

backwards compatibility.

1049

00:38:45,130 --> 00:38:46,460

And that is very different.

1050

00:38:46,460 --> 00:38:48,280

And also you got to be very

1051

00:38:48,280 --> 00:38:51,040
cognisant of every
programming language

1052
00:38:51,040 --> 00:38:52,875
sort of has a gestalt to it,

1053
00:38:52,875 --> 00:38:54,830
if it's an object doing
a programming language,

1054
00:38:54,830 --> 00:38:56,720
you can move it towards
funtional program

1055
00:38:56,720 --> 00:38:58,905
but you can't make it into
a functional program.

1056
00:38:58,905 --> 00:39:02,350
Like where immutability
is first. So there's-.

1057

00:39:02,350 --> 00:39:04,000

>> There's iterators, but yeah.

1058

00:39:04,000 --> 00:39:05,280

>> Right, there are

certain things where you just

1059

00:39:05,280 --> 00:39:06,740

got to go, "No,

1060

00:39:06,740 --> 00:39:10,545

if we do this that'll just

give people two ways of doing

1061

00:39:10,545 --> 00:39:13,490

the same thing and now there

will forever be confusion

1062

00:39:13,490 --> 00:39:16,320

about which one it is

that I'm supposed to use.

1063

00:39:16,320 --> 00:39:18,020

So, we can't do that."

1064

00:39:18,020 --> 00:39:20,720

So, after 1.0 you

have to learn when to

1065

00:39:20,720 --> 00:39:23,485

say no and save it

for the next one,

1066

00:39:23,485 --> 00:39:25,730

which is hard because there

are things you want to do

1067

00:39:25,730 --> 00:39:28,270

but this is not

the right place to do it.

1068

00:39:28,270 --> 00:39:30,085
And you could sink the
ship if you did and then,

1069
00:39:30,085 --> 00:39:31,720
I think you learn also to

1070
00:39:31,720 --> 00:39:35,330
withstand the pressures
of marketing.

1071
00:39:35,330 --> 00:39:39,554
I can't tell you how many mails
I have written to defend

1072
00:39:39,554 --> 00:39:44,150
our decision not to put
XML literals into C sharp.

1073
00:39:44,150 --> 00:39:45,770
I literally spend a year

1074

00:39:45,770 --> 00:39:47,750

writing that email

every two weeks.

1075

00:39:47,750 --> 00:39:49,305

And Visual Basic did put

1076

00:39:49,305 --> 00:39:51,615

XML there and I was

always like "Listen,

1077

00:39:51,615 --> 00:39:53,110

there are so many different ways

1078

00:39:53,110 --> 00:39:54,255

you can write this markup.

1079

00:39:54,255 --> 00:39:59,165

If we put it in an XML sinks

then we sink along with it."

1080

00:39:59,165 --> 00:40:01,050

Can you imagine if we'd had

1081

00:40:01,050 --> 00:40:03,090

XML literals in C

sharp. Now what-.

1082

00:40:03,090 --> 00:40:04,270

>> I can't even imagine.

1083

00:40:04,270 --> 00:40:06,690

>> And it's sometimes

you just got to stick to

1084

00:40:06,690 --> 00:40:09,430

your guns and the face

of adversity and go "No,

1085

00:40:09,430 --> 00:40:10,980

this does not belong here,

1086

00:40:10,980 --> 00:40:12,650

here is what I know, I understand

1087

00:40:12,650 --> 00:40:14,240

how with the vantage point of

1088

00:40:14,240 --> 00:40:16,300

today it might look

like it does belong

1089

00:40:16,300 --> 00:40:18,665

here but it doesn't

belong here, long term."

1090

00:40:18,665 --> 00:40:19,355

>>Yeah.

1091

00:40:19,355 --> 00:40:20,570

>> So, there are lots of

1092

00:40:20,570 --> 00:40:23,345
interesting things that you
don't realize up front.

1093
00:40:23,345 --> 00:40:24,625
>> I've always thought that

1094
00:40:24,625 --> 00:40:27,730
the better design programming
languages were the ones

1095
00:40:27,730 --> 00:40:30,950
where there was
some coherent philosophy

1096
00:40:30,950 --> 00:40:34,135
that was developer
oriented up front.

1097
00:40:34,135 --> 00:40:36,080
Pearl, which in many ways this

1098

00:40:36,080 --> 00:40:39,960

is I've written a lot of

bad Pearl code in my time.

1099

00:40:39,960 --> 00:40:42,235

It's really easy to

do and it's my fault,

1100

00:40:42,235 --> 00:40:45,590

not language's fault but it

had a lot of good stuff.

1101

00:40:45,590 --> 00:40:48,710

Like Larry Wall

the creator of Pearl had

1102

00:40:48,710 --> 00:40:50,865

these philosophical tenets like

1103

00:40:50,865 --> 00:40:54,020

make the easy things easy and

the hard things possible.

1104

00:40:54,020 --> 00:40:56,110

I remember hearing Fred Brooks at

1105

00:40:56,110 --> 00:40:58,020

one point say one

of the best things

1106

00:40:58,020 --> 00:40:59,820

you can do in language design

is to figure out

1107

00:40:59,820 --> 00:41:01,880

the things that you're not

going to let people say.

1108

00:41:01,880 --> 00:41:04,360

Because that's where

mistakes happen.

1109

00:41:04,360 --> 00:41:06,420
So it's always
fascinating to me what

1110
00:41:06,420 --> 00:41:08,600
these philosophical principles
are because there's

1111
00:41:08,600 --> 00:41:11,385
no one right way to design
a programming language.

1112
00:41:11,385 --> 00:41:13,765
>> Right, but there are
lots of wrong ways,

1113
00:41:13,765 --> 00:41:16,330
and I think that one thing
that people tend to

1114
00:41:16,330 --> 00:41:17,580
forget too about programming

1115

00:41:17,580 --> 00:41:18,895

languages is well

a couple of things.

1116

00:41:18,895 --> 00:41:21,139

First of all, it's fascinating

1117

00:41:21,139 --> 00:41:23,270

to look at how slowly

they evolve and

1118

00:41:23,270 --> 00:41:25,650

how similar the

programming languages

1119

00:41:25,650 --> 00:41:28,420

of today are to the ones

we use 50 years ago.

1120

00:41:28,420 --> 00:41:30,685

It's really quite frightening.

1121

00:41:30,685 --> 00:41:33,900

I mean like Pascal

was invented in 1974,

1122

00:41:33,900 --> 00:41:36,270

so we're talking about

like 40 some years

1123

00:41:36,270 --> 00:41:39,650

here and it doesn't really

look all that different.

1124

00:41:39,650 --> 00:41:41,290

Okay, we've had a few, okay,

1125

00:41:41,290 --> 00:41:42,685

object orientation but then

1126

00:41:42,685 --> 00:41:44,460

Pascal looks a bit

1127

00:41:44,460 --> 00:41:45,850

like a functional

programming language,

1128

00:41:45,850 --> 00:41:46,930

it doesn't have pattern matching,

1129

00:41:46,930 --> 00:41:49,390

it doesn't have lambdas or

whatever but it has some of it

1130

00:41:49,390 --> 00:41:52,180

like functions and data

being separated and anyway.

1131

00:41:52,180 --> 00:41:53,960

So that's one thing

you come to realize.

1132

00:41:53,960 --> 00:41:57,780
I think another one is
that it's very easy to

1133
00:41:57,780 --> 00:41:59,680
focus on one shiny thing

1134
00:41:59,680 --> 00:42:02,165
that you're programming
language is going to do.

1135
00:42:02,165 --> 00:42:04,515
But then you always forget about

1136
00:42:04,515 --> 00:42:06,740
the 90 something other percent

1137
00:42:06,740 --> 00:42:09,240
that every programming
language has to have.

1138

00:42:09,240 --> 00:42:12,100
And that 90 other
something percent keeps

1139
00:42:12,100 --> 00:42:15,190
getting bigger and bigger now
because people need tools,

1140
00:42:15,190 --> 00:42:18,540
they need IDEs, they need
statement completion,

1141
00:42:18,540 --> 00:42:19,980
they need code navigation,

1142
00:42:19,980 --> 00:42:21,455
they need blah, blah, blah.

1143
00:42:21,455 --> 00:42:22,950
And so whenever you're like,

1144

00:42:22,950 --> 00:42:25,480

"Oh I got to have

my little DSL like thingy

1145

00:42:25,480 --> 00:42:27,240

here just for configuring

1146

00:42:27,240 --> 00:42:29,170

this thing, oh

wouldn't it be cool."

1147

00:42:29,170 --> 00:42:30,680

Well, are you going to build

1148

00:42:30,680 --> 00:42:32,105

an ID VoIP or you're

going to build two?

1149

00:42:32,105 --> 00:42:33,700

"Oh no, they're never going to

1150

00:42:33,700 --> 00:42:35,980
get big enough for
these programs."

1151
00:42:35,980 --> 00:42:39,350
Well, every configuration
file grows up to be

1152
00:42:39,350 --> 00:42:41,180
its own crappy
programming language

1153
00:42:41,180 --> 00:42:43,055
eventually and if
you don't watch out.

1154
00:42:43,055 --> 00:42:44,715
And then they get bigger

1155
00:42:44,715 --> 00:42:46,380
and bigger and bigger
and now you need

1156

00:42:46,380 --> 00:42:50,310

tools and now there we are

repeating history again.

1157

00:42:50,310 --> 00:42:53,640

>> Some of the hardest things

that I've had to do as

1158

00:42:53,640 --> 00:42:55,620

someone leading teams

of engineers

1159

00:42:55,620 --> 00:42:57,785

is to be the bad guy

coming and saying,

1160

00:42:57,785 --> 00:43:01,390

"Okay, we're not going to use

that shiny new language."

1161

00:43:01,390 --> 00:43:03,715
And it's precisely for
that reason when you're

1162
00:43:03,715 --> 00:43:05,970
figuring out
what development tools

1163
00:43:05,970 --> 00:43:08,370
you're going to
use to empower all

1164
00:43:08,370 --> 00:43:11,160
of your engineers to
build great things,

1165
00:43:11,160 --> 00:43:13,530
you have to make
choices because all of

1166
00:43:13,530 --> 00:43:16,295
the apparatus that you build

around them to support them,

1167

00:43:16,295 --> 00:43:17,530

what's the build system

1168

00:43:17,530 --> 00:43:18,850

going to look like

when you've got

1169

00:43:18,850 --> 00:43:20,040

hundreds of millions of lines of

1170

00:43:20,040 --> 00:43:22,055

code and thousands of engineers?

1171

00:43:22,055 --> 00:43:24,160

How do you do

continuous integration

1172

00:43:24,160 --> 00:43:25,830

and continuous deployment?

1173

00:43:25,830 --> 00:43:28,170

What is your testing

regime look like?

1174

00:43:28,170 --> 00:43:29,680

So you build all this automation

1175

00:43:29,680 --> 00:43:31,920

that sits around the

programming language,

1176

00:43:31,920 --> 00:43:33,570

the engineers can sometimes

1177

00:43:33,570 --> 00:43:35,210

have this notion in

their head it's like, "Oh,

1178

00:43:35,210 --> 00:43:36,495

it can't be that hard,

1179

00:43:36,495 --> 00:43:38,210

I'll just write my one thing in

1180

00:43:38,210 --> 00:43:40,480

this and it'll be okay." And then-

1181

00:43:40,480 --> 00:43:41,880

>> That can happen to you,

1182

00:43:41,880 --> 00:43:44,300

it particularly the

higher you rise

1183

00:43:44,300 --> 00:43:47,140

in the engineering ranks and

the less code you write,

1184

00:43:47,140 --> 00:43:51,790

the more possible you think

the impossible is, right?

1185

00:43:52,550 --> 00:43:56,875

You become an architecture

astronaut where surely,

1186

00:43:56,875 --> 00:43:58,820

anything can be done here, right?

1187

00:43:58,820 --> 00:44:00,710

Look at this diagram, I

just drew it for you.

1188

00:44:00,710 --> 00:44:02,020

Just go and make it so, right?

1189

00:44:02,020 --> 00:44:04,935

No, coding is hard and

it continues to be hard.

1190

00:44:04,935 --> 00:44:06,460

Code gets bigger and

bigger and bigger

1191

00:44:06,460 --> 00:44:08,270

but our brains are not

getting any bigger,

1192

00:44:08,270 --> 00:44:11,335

and this is largely

a brain exercise.

1193

00:44:11,335 --> 00:44:12,025

>> Yeah.

1194

00:44:12,025 --> 00:44:17,320

>> So one of the things

that I end up talking with

1195

00:44:17,320 --> 00:44:19,830

fellow engineers about a lot over

1196

00:44:19,830 --> 00:44:21,950

the past several

years is where we are

1197

00:44:21,950 --> 00:44:24,190

now as a discipline,

1198

00:44:24,190 --> 00:44:26,705

as a profession, compared

to where we were before.

1199

00:44:26,705 --> 00:44:27,930

So, in some ways,

1200

00:44:27,930 --> 00:44:29,830

the programming languages from

1201

00:44:29,830 --> 00:44:32,840

a language perspective aren't

1202

00:44:32,840 --> 00:44:35,270

that much different

now than they were.

1203

00:44:35,270 --> 00:44:38,045

But you do have

these crazy things like

1204

00:44:38,045 --> 00:44:41,355

cloud computing and

open source software where,

1205

00:44:41,355 --> 00:44:43,150

I can't tell you how many times,

1206

00:44:43,150 --> 00:44:44,470

when I was a young engineer I

1207

00:44:44,470 --> 00:44:47,870

re-implemented chain

hashing implementation.

1208

00:44:47,870 --> 00:44:49,200

It's like I could

probably still write

1209

00:44:49,200 --> 00:44:50,800

the code just out

of finger memory.

1210

00:44:50,800 --> 00:44:51,415

>> Right. Right.

1211

00:44:51,415 --> 00:44:52,720

>> But like so much of this stuff

1212

00:44:52,720 --> 00:44:53,995

is accounted for you, so,

1213

00:44:53,995 --> 00:44:57,050

you've got really

robust SDK's and like

1214

00:44:57,050 --> 00:44:59,165

a standard class library for

1215

00:44:59,165 --> 00:45:00,660

programming languages

and then you've got

1216

00:45:00,660 --> 00:45:02,255

like all of this infrastructure.

1217

00:45:02,255 --> 00:45:04,810

What you can sit

down in a weekend

1218

00:45:04,810 --> 00:45:07,635

and write in terms

of internet service

1219

00:45:07,635 --> 00:45:10,085

application, compile it,

1220

00:45:10,085 --> 00:45:12,635

press long button and

ship it to a cloud,

1221

00:45:12,635 --> 00:45:15,400

then expose it to

the world. It's like crazy.

1222

00:45:15,400 --> 00:45:17,170

>> No. It's phenomenal

what's happened here.

1223

00:45:17,170 --> 00:45:20,500

I think a lot of it

is like- opensource

1224

00:45:20,500 --> 00:45:23,995

has done a tremendous

amount to shorten

1225

00:45:23,995 --> 00:45:27,550

the distance between individual

programmers and between

1226

00:45:27,550 --> 00:45:29,610

users of someone implementing

1227

00:45:29,610 --> 00:45:31,280

a tool and someone using a tool.

1228

00:45:31,280 --> 00:45:32,700

We're this close. I mean,

1229

00:45:32,700 --> 00:45:34,630

I feel it on my body

every day, right?

1230

00:45:34,630 --> 00:45:36,610

I sit on the GitHub issue tracker

1231

00:45:36,610 --> 00:45:37,885

on TypeScript and then I'm

1232

00:45:37,885 --> 00:45:39,590
talking to the very people who

1233
00:45:39,590 --> 00:45:41,420
are using it and if
they report a bug,

1234
00:45:41,420 --> 00:45:42,780
we can have it turned around and

1235
00:45:42,780 --> 00:45:45,200
have it in the nightly
bill that evening,

1236
00:45:45,200 --> 00:45:47,650
which that used to
be a two-year cycle.

1237
00:45:47,650 --> 00:45:49,820
We have done it like
old proprietary style.

1238

00:45:49,820 --> 00:45:52,505

So, the pace of advance

1239

00:45:52,505 --> 00:45:55,550

there and iteration is just

gone up so much. That is-

1240

00:45:55,550 --> 00:45:57,780

>> It's also that you doing

it in the open, right?

1241

00:45:57,780 --> 00:45:57,960

>> Sure.

1242

00:45:57,960 --> 00:45:59,780

>> Like so many people

are making so much of

1243

00:45:59,780 --> 00:46:02,380

this stuff available

for low or no cost.

1244

00:46:02,380 --> 00:46:02,825

>> Yeah.

1245

00:46:02,825 --> 00:46:04,950

>> That is still

a strange phenomenon,

1246

00:46:04,950 --> 00:46:06,830

because ultimately,

someone's got to pay, right?

1247

00:46:06,830 --> 00:46:07,010

>> Right.

1248

00:46:07,010 --> 00:46:09,605

>> I don't know where that's

going all land eventually,

1249

00:46:09,605 --> 00:46:12,440

but I have to say I do love

the process of doing it.

1250

00:46:12,440 --> 00:46:14,420

It's sort of like a return to

1251

00:46:14,420 --> 00:46:17,655

the things that really

matter about programming.

1252

00:46:17,655 --> 00:46:19,640

I mean our team, it's

not a big team but

1253

00:46:19,640 --> 00:46:21,600

it allows us to be

craftsmen and it

1254

00:46:21,600 --> 00:46:25,320

allows us to really just

continue to iterate. I was always

1255

00:46:25,320 --> 00:46:27,560

a great fan of- you

don't just build

1256

00:46:27,560 --> 00:46:28,950

a product and throw it out there

1257

00:46:28,950 --> 00:46:30,490

and then you move on

to a whole new thing.

1258

00:46:30,490 --> 00:46:33,345

No, you got to come

back and polish it.

1259

00:46:33,345 --> 00:46:36,170

Then iterate on it and make

it better and make it better.

1260

00:46:36,170 --> 00:46:38,880

It doesn't get beautiful

just out of the box.

1261

00:46:38,880 --> 00:46:41,870

It gets beautiful over

a period of years of

1262

00:46:41,870 --> 00:46:45,535

constant iteration and care.

Do you know what I mean?

1263

00:46:45,535 --> 00:46:46,030

>> Yes.

1264

00:46:46,030 --> 00:46:47,890

>> That's something

that this ecosystem

1265

00:46:47,890 --> 00:46:49,100

allows us to do at

1266

00:46:49,100 --> 00:46:52,320

such a phenomenal pace

that it's just beautiful.

1267

00:46:52,320 --> 00:46:54,910

We could have never done

what we're doing now with

1268

00:46:54,910 --> 00:46:55,980

Visual Studio Code and

1269

00:46:55,980 --> 00:46:58,400

TypeScript in

the old proprietary world.

1270

00:46:58,400 --> 00:46:59,920

It simply would have taken

1271

00:46:59,920 --> 00:47:01,680

100 years to get that amount of

1272

00:47:01,680 --> 00:47:03,690

feedback incorporated

on the kinds

1273

00:47:03,690 --> 00:47:05,250

of cycles that we

were on back there.

1274

00:47:05,250 --> 00:47:05,510

>> Right.

1275

00:47:05,510 --> 00:47:07,040

>> So that's great.

>> That's awesome.

1276

00:47:07,040 --> 00:47:07,290

>> Yeah.

1277

00:47:07,290 --> 00:47:09,615

>> I think maybe that's

a good point to stop.

1278

00:47:09,615 --> 00:47:11,590

We've gone from core memories

1279

00:47:11,590 --> 00:47:14,880

to the sort of near infinite

1280

00:47:14,880 --> 00:47:18,350

rate of change in open source software.

1281

00:47:18,350 --> 00:47:19,880

So, thank you so much.

1282

00:47:19,880 --> 00:47:20,910

>> All my pleasure.

It was great, fun.

1283

00:47:20,910 --> 00:47:22,000

>> It was awesome, so fun.

1284

00:47:22,000 --> 00:47:23,870

>> Fun talk, yeah.

1285

00:47:25,020 --> 00:47:27,630

>> Thanks for joining me

from Behind the Tech.

1286

00:47:27,630 --> 00:47:30,005

It was a lot of fun

speaking with Anders.

1287

00:47:30,005 --> 00:47:31,550

I had some surprises in there,

1288

00:47:31,550 --> 00:47:34,385

like I didn't realize

that he had written

1289

00:47:34,385 --> 00:47:36,769

a precursor to Turbo Pascal

1290

00:47:36,769 --> 00:47:39,220

while he was still in university.

1291

00:47:39,220 --> 00:47:41,240

That may very well be

1292

00:47:41,240 --> 00:47:43,260

the first integrated

development environment

1293

00:47:43,260 --> 00:47:45,285

in the entire world.

1294

00:47:45,285 --> 00:47:48,120

And he's so modest about it,

like, "I never thought of it."

1295

00:47:48,120 --> 00:47:49,600

It's like God Almighty,

1296

00:47:49,600 --> 00:47:51,230

this is one of the more

1297

00:47:51,230 --> 00:47:53,490
important breakthroughs
from point of view

1298
00:47:53,490 --> 00:47:55,085
of the developer that has

1299
00:47:55,085 --> 00:47:57,570
happened in the
history of computing.

1300
00:47:57,570 --> 00:48:00,740
So, many thanks to Anders
for being on the program.

1301
00:48:00,740 --> 00:48:02,830
>> Next time on Behind the Tech,

1302
00:48:02,830 --> 00:48:06,135
we'll hear from Alice Steinglass
who heads Code.org.

1303

00:48:06,135 --> 00:48:07,520

We'll hear about how

she's working to

1304

00:48:07,520 --> 00:48:08,890

make sure every kid has

1305

00:48:08,890 --> 00:48:09,910

the opportunity to take

1306

00:48:09,910 --> 00:48:11,860

computer science

classes in school.

1307

00:48:11,860 --> 00:48:14,330

Be sure to tell your friends

about our new podcast,

1308

00:48:14,330 --> 00:48:18,390

Behind the Tech, and to

subscribe. See you next time.

Behind the Tech with Kevin Scott

EP 02 – Anders Hejlsberg: A craftsman of computer language