

Research Project Workshop: Data

The following transcript is from a field study investigating the progress of an organisation adopting the systems and practices of Extreme Programming (XP) for commercial software development. Interviews were conducted in 2001 over the period from April through to June. Interviews and discussions were recorded on audiotape and transcribed afterwards. The participant company, project teams, individuals and other features are anonymised to remove identifying references.

The group discussion transcript indicates speaker identity (anonymised) in bold text. The investigation employed semi-structured interviews, guided by the research questions below:

Table A: Research Questions

- ⊗ Has XP been perceived as successful?
- ⊗ Has XP changed or altered the way you work?
- ⊗ Has XP changed attitudes in the organisation?
- ⊗ How has your roll-out of XP been managed and communicated?
- ⊗ Do you think XP will reach a steady state or is it an on-going process?
- ⊗ What systems do you need in place to support your use of XP?
- ⊗ Is XP more difficult to use than the previous approach?
- ⊗ Would you be happy to return to the previous approach?
- ⊗ What if anything would you bring from XP onto a new project?
- ⊗ What do you understand by...?
 - Coding standards
 - Small Releases
 - Metaphor
 - Simple Design
 - Testing
 - Continuous Integration
 - The Planning Game
 - Pair Programming
 - Collective Ownership
 - 40 hour week
 - On-site customer
 - Refactoring
- ⊗ Are there any general observations you'd like to make about XP?

TEXT: Group discussion

- Engineering Manager** I was pleasantly surprised that you chose to implement code review via pair programming (.). What about the other areas of XP (?) How are **we** doing (?) What about the other XP practices and the infrastructure implied (.)
- Product Manager** I know XP is working extremely well in engineering (.). I would often hear that people here (.). not in engineering (.). concerned and wondering if XP is really working (?) Particularly in a customer facing situation like Customer B (.)
- Test Engineer 030** Sometimes I wonder if Customer B was the right customer to benchmark XP on (.). because XP came along halfway through (.). I don't know how much of Customer B was XP (.). How many of the core practices did **we** actually implement during the Customer B development (?)
- Engineering Manager** Worth looking at (.). The core practices haven't been rolled out at this time (.). we're still focused on building the infrastructure to allow us to do XP (.). What **we** have been doing is transition to the new source code control system and fine-tune our unit testing tools and business testing tools (.). **We've** also put some of the infrastructure in place around communication (.). Customer B features were added to the Wiki (.). and adopted the TaskBoard for macro tracking (.). StoryCards for discrete details of any particular development (.).
- IdealDays for estimation purposes (.). Feature and Bug queues to facilitate volunteering (.). From the test team perspective is XP appropriate (?)
- Test Engineer 030** From our point of view the biggest problem that we'd encounter is not knowing what we're going to get until **we** get it (.). **we** have a way of planning that allows us to free up resources when **we** expect something to come in (.). If we're going to take Customer B as an example of the delivery of XP (.). very often **we** only got a partial implementation and **we** find for example that where previously **we** could have done 15 days of testing at the end of getting something complete (.). now **we** are fire-fighting to get it to a certain stage because we've only gotten partial deliveries in releases (.). I don't know how much of that is time pressure (.). how much of that is XP (.). that's very difficult to tease out (.). Definitely **we** wasted a lot of time at the beginning waiting for something to test then when **we** got it (.). it was only half complete and **we** weren't getting any further in our test plans (.).
- Engineering Manager** Did you find that you had more input to the implementation (?)

Test Engineer 030 Yes (.) definitely **we** did (.) definitely because of the too and froing that **we** did (.) but that's because **we** were looking at something that wasn't ready for system test (.) Or system tested (.)

45 **Engineering Manager** Was that a good thing or a bad thing (.) that you had input into the implementation (.)

Test Engineer 030 It was definitely a good thing (.) but overall **I** think some of our effort was wasted (.)

50 **S/W Engineer 066** **I** would say that some of your effort was diluted from an early stage (.) primarily because there was huge time pressure for the Project B functional work to be done by a particular date (.) One area that **we** didn't factor was gluing all the branches together under source code control (.) One engineer took a week to get that build (.) at the end of that week you were hammering on the door to get a build into test (.) which
55 wasn't smoke tested like **we** had done in Project R (.) where **we** did tremendous work in testing builds before you guys got them (.) But that didn't happen cause you guys were banging on the door (.) **I** don't know whose fault that is (.)

60 **Test Engineer 030** But that's why I'm wondering is that XP or is that XP's problem (.)

Product Manager That's just time pressure

65 **S/W Engineer 066** It's time pressure on product functionality (.) you had to have something by X date (.) and stuff that was to be put into the product was taking longer than that date (.) So there was no actual smoke testing done this time (.) So you were doing the smoke testing this time (.) Hopefully in the newer iteration the work that engineering is doing on automation subsystem (.) some of the tests (.) not all of them (.) that is where XP will come to the fore (.) where automation is the key to XP but its not there yet (.)

70 **Test Engineer 030** That's why I'm wondering (.) is this XP that is the problem or are there other factors coming into it (.)

75 **S/W Engineer 066** Put it like this (.) the entire organisation can't become XP overnight (.) **we** will take 5 or 6 months to get into the XP style of doing things (.) **we** won't be doing all the core practices we'll probably adopt most of them (.)

80 **S/W Engineer 042** But even Project B (.) it was "we need this functionality and we need it by this date" (.) its not XP (.) You don't do this much by this date (.) **we** were trying to squeeze as much functionality in by a particular date (.) and what happens is what always happens is that you get functionality that is not properly tested (.) you rush to get it in by a particular date and you haven't spent as much time doing full design and testing that you might have (.) thinking it through the whole way and test end up getting stuff that is not fully complete (.)

85 **Engineering Manager** (inaudible) think our starting point is an acceptance that **we** are not currently doing XP (.) It's a horizon **we** are heading towards (.)

90 **S/W Engineer 100** **I** would say **we** are definitely not doing XP (.) The core practices is testing and pair programming (.) They are the time savers ultimately (.) that's why **we** aren't saving time yet (.) Pair programming **we** have to prove but automated testing **we** know produces massive savings in the long run (.)

S/W Engineer 112 Out build and test system needs to be more visible too (.) that'll provide huge benefits (.)

95 **Product Manager** These conflicts that go on with a live customer situation (.) would they be alleviated when you are using XP and the customer knows about it (.) **I** know what you are saying there where a customer says **I** need this functionality by this date (.) but that's the reality of life with any customer (.)

S/W Engineer 066 That's what's going to happen with Project A (.)

100 **S/W Engineer 100** **I** wonder if that's necessarily true (.) with the waterfall model (.) when any new model is introduced there an exercise there in training customers to understand the model (.) to buy into the model (.) and follow the model (.) So waterfall for example they had to give all the requirements up front (.) then they get something which met those
105 requirements (.) and that's the way it worked (.) And now that's probably the way most customers think in terms (.) They give all the requirements and then they get something (.) And now **we** have another model (.) and **I** think there is an education of customers required *yet* again (.) And that's something that will happen very slowly (.)

110 **Product Manager** So what you are saying is that the customer should be involved in this whole approach... (?)

S/W Engineer 100 Absolutely (.) Yes

Product Manager And **I** can see the situation where XP is used as a positive selling process (.)

115 **S/W Engineer 100** Exactly (.) because if they've used approaches like waterfall before then they know that they've given their requirements and 9 months later they've got something that probably did not meet the mark (.)

120 **Product Manager** **I** still think that you're still going to be up in the situation with a customer where they say like for project A where they've got to go live by a particular date and **we** have to have this and **we** have to have that (.) You can educate them to a degree but the thing is always the case that they will push the dates (.)

125 **Engineering Manager** XP was derived from working in that particular bespoke or consulting environment (.) where they recognised that quite often what the customer says they wanted isn't what they need (.) and its

only by seeing intermediate releases (.) partial functionality that the need can be met (.) That's what XP allows for (.) enabling this fundamental relationship between the business (.) customer or end-user and the developer to exist (.) be healthy (.) for real communication to happen (.)

130 **S/W Engineer 124** The key point there is that everyone has a customer "on tap" (.) where that's a real end-user or it's someone within the organisation that represents the customer like the Product Managers or Test People (.) **I** think that's very important because you're getting input from customers (.) until a certain cut-off date probably but they can still give you input where you can say this is what **I** have now (.) you can tell them and they say "can we refine it a little bit this way and that way" (.) **I** think that works very well (.) You're always getting input so you don't go off down a certain road (.) at a stage (.) before when you've gone so far you find you've got so many changes to make to get back to what was needed (.) This way you've always getting customer input (.)

140 **S/W Engineer 065** The people that were speaking at the XP SIG (.) they were saying it was a big deal for them (.) that they could go to their customers and the customers felt that they had involvement in the product (.) that they could specify at several stages (.) they felt that that was a bit of a selling point (.)

150 **Test Engineer 030** **I** can't speak for everyone (.) from a testing point of view (.) the only thing that **I** think **we** fall down on is a realisation on the amount of preparation that gets put into to get ready to test (.) be it a new feature or a fix (.) **we** need input in order to plan (.) The idea that **we** don't know what we're going to get (.) that's the biggest worry **we** have (.) In a way **we** really wouldn't mind if something was even a partial implementation if **we** knew exactly how far that was going to go (.) And that wasn't going to block some other testing of some other testing of other features further down the way cause **we** were waiting for it to go further (.) So it's the whole planning cycle (.) **we** maybe need to work on a bit in terms of us getting deliverables (.) And the fact that you're delivering by XP doesn't even really make a difference to use (.) but **we** do need to be able to plan our tests efficiently and effectively (.) Before the day **we** actually get the features working and that's the biggest thing

160 **S/W Engineer 065** So did you used to get that before **we** started to move to XP (.)

Test Engineer 030 ... **we** were working on that (.) that's why I'm not sure if this is particularly to do with XP (.) **I** think for project R **we** where hitting those issues due to time pressures (.)

165 **S/W Engineer 065** Is that fact that you're getting notes continuously updated with new fixes or functionality (.) does that help you?

Test Engineer 030 Yes (.) it does (.) absolutely; because **we** know exactly what was fixed in a build (.) but that's something **we** were working on in project R (.) so it's not something that came with XP (.)

170 **we** were talking about this whole serialisation problem (.) that might help with the "Spike" sessions (.)

175 **S/W Engineer 066** **I** think a lot of the aspects that you (the Test People) need to ramp up on are the same as in here (engineering) (.) automation is the name of the game (.) for turning builds around (.) Because XP is designed to give you smaller changes in succession (.) Our product is so large (.) for us to do a bug fix (.) you can't afford to do a full system test for each small bug fix or release (.) You were hoping in the case of Customer B that you could take each build with one small fix and test that (.) expecting other functionality to remain unaffected (.) but you need and **we** need as well (.) to make sure that you can do as much automated testing as possible (.) **we** won't be doing another project R where you received single builds in cycles and one or two complete features (.) You can't afford to be in that mind set any more (.) you've got to get out of it (.) it won't work in Test and it definitely won't work here (.)

185 **Test Engineer 030** **I** would absolutely agree with that (.)

190 **S/W Engineer 066** And you've got to be able to go to a machine (.) say a test rig (.) and restore it to a clean start-up basis (.) Where you've got all your test data and scenarios (.) you've got to get to that state in an instant (.) That will save hours or days (.) where this driver won't work with this database (.) you can't connect to database A (.) or you won't have to look for the original CDs (.) ask who's got that copy and where are the patches or system service packs (.) Its things like that that enable you to work faster (.) if you don't do that you won't be able to do XP (.)

195 **S/W Engineer 042** Are you going to the board (.) to see what is ready or fixed (?) **I** imagine part of this is to know (.) as part of a build (.) to know what is fixed or what is going in (.) at our end (.) is the board not detailed enough to show what's going in (.)

200 **Test Engineer 030** **I** think by the time it's gotten onto the board (.) it's too late for test planning (.) **I** do agree that automation is the way to go (.) and **we** need a regression suite to say that none of the core functionality has changed (.) But there will be new features and it's the new features and new things coming onto the board that **we** need to be able to plan for (.) **we** need some information before it even starts (.) that's the key (.)

205 **S/W Engineer 101** But **we** can't provide that detail before the work is done (.) it's got to be done at the same time or close to the time **we** accept the requirements (.) So you've got to be working in parallel with the development (.) building your test plans to match the implementation and requirements (.) because the requirements change too as **we** understand the need (.)

210 **Engineering Manager** So we've been toying with the idea of a "Spike" (.) a series of meetings where a customer "story" is fleshed out with a software functional requirements and possibly design (.) that the test plan should be built there too (.) at the same meeting (.) we can't continue to attempt to serialise the requirements (.) design (.) coding and test stages of

215 conventional development (.) Anyway this isn't the way I've observed you as a team really working (.) What is the way you really work (?) As a team (.) how did you really deliver the latest set of features (?)

S/W Engineer 066 "Feature Y" is a classic (.) we had a specification from the standards group for data we had to support (.) So we had no need to build up our own written requirements (.) we had a stand-up meeting to flesh out the design (.) Basically we did a "Spike" to push progress (.)

S/W Engineer 100 That works because the specification is a detailed requirement in itself (.) we had that requirement specification already (.)

Engineering Manager Take a requirement like "we need a special format" (.) I think that statement is enough to start (.) so long as the person that requires this (.) so long as the person that defined that statement is there to correct the stand-up design and annotate their requirements somewhat (.)

S/W Engineer 100 Well that is in fact the way we work (.) there's a kind of bartering going on between the implementer and the specifier as such (.)

230 to arrive at a consensus (.)

Product Manager Is that what is known as RAD (?)

S/W Engineer 113 A key part of XP is to reduce the requirement risks (.) so before going to the next stage the requirement needs to be evaluated in detail (.) by a combination of Product Manager and Engineer (.) I think the main purpose of XP is to reduce the risk of requirements being understood (.) not that the requirements are wrong (.) but that we discuss and understand the requirement better in XP (.)

S/W Engineer 042 we need to be engaged in understanding the requirement; otherwise the requirement may not be a solid basis on which to develop (.)

240 **Engineering Manager** One of you made an observation about this change that has come into software development; it's the potential for automation (.) The profession has taken a long time to respond (.) and realise the fundamental change that the capability to run the tests automatically (.) regularly (.) The implication of Automated testing is where XP started (.)

245 The other ways (.) RAD (.) Spiral (.) were adaptations from our earlier practices (.) from the days where the scope of an entire operating system was within the span of a single person (.) whole environments fitted onto single floppy disks (.) so teams of testers could realistically provide

250 complete product coverage in a reasonable period of time (.) So the huge size of our new application environments (.) can only be answered by automated test (.) and understood when you approach it through XP (.)

Test Engineer 030 What I'm wondering at (.) we've seen how XP fits a development model (.) but have we seen any companies that have taken XP

255 across the board (?) What kind of practices do they do (?) Its putting flesh on the bones of these ideas that we need to look at (.)

Engineering Manager I think you'll find that XP says nothing about corporate governance and operations (.) interdivisional communications in this instance (.) we could make it say something (.) but XP doesn't say that as it is presented (.)

260 **Test Engineer 030** Surely there are other companies who have taken XP on (.) even if it is only within their development environment (.) a company like Company Z (.) uses a completely different test structure (.) Where they have a tester sitting on a team of pair programmers and that goes through a cycle before it ever gets to test (.) Do know exactly in other companies that are doing XP (?)

265 **Engineering Manager** They are grappling with the same issues (.) XP does seem to put stresses on other parts of the organisation (.) there is also this idea of XP contracts (.) a commercial approach to XP (.) However this XP project is only concerned with software development (.) What is your concern (?)

270 **Test Engineer 030** I would like to understand best practice for how to get the job done (.) presuming that development are taking XP on board (.) There's no point in us working in a way that is delaying test and thereby delaying the end product (.) So if there are models which show us how to work along side development (.) to do this the best we can (.)

275 **S/W Engineer 124** I'm just wondering (.) the key thing seems to be for automated testing to be implemented (.) You'd run a suite of tests and that'd be it (.) is there a case to made for us to put a couple of engineers on it continuously (.) working on our source base automating tests full time (?) Two engineers or rotate it whatever (?) There's a lot of work involved (.) it's a huge codebase (.)

280 **Test Engineer 030** The sort of tests that I'd want is to have a regression suite that goes way beyond GUI testing (.) but that's not the end of the world (.) that's just the beginning (.) that's not an end of itself (.) So maybe we do need to look at hiring people specifically for that purpose (?) I do believe that's the way we have to go and it would certainly give a lot of confidence for regression testing (.)

285 **S/W Engineer 124** You're getting builds for Customer B now (.) based on partially implemented functionality (.) but haven't been smoke tested (.) As mentioned already (.) when engineering's automated tests are ready (.) then you've got the confidence that its been through smoke testing and you can focus on functionality that's new (.) and you should be able to go faster just testing the new (.) with confidence high (.)

290

295 **Test Engineer 030** Especially when **we** go cross platform as well (.) you've got 6 weeks system testing for each platform (.)

Engineering Manager There are books on XP (.) but they're not "how-to's"

Product Manager Maybe a lot of people don't understand the benefits to them of XP (?) What about the test engineers (.) it's up to us to sell the idea to each of you (.) to convince you that XP has something to offer (.) marketing (.) sales (.) **I** don't think it's a problem with XP (.) It's a shame in some ways that someone from marketing and sales didn't come to this meeting (.)

300

305 **Engineering Manager** **we** come back to the realisation that **we** still aren't doing XP (yet) (.)

S/W Engineer 112 Is pair programming accepted now (?) Has everybody bought into that (?) It's the one practice that I'm not sold on (.) **I** can see benefits (.) however (.) I'm nervous about it (.) things like personalities (.) things like people's programming habits (.) and people being people (.) It can be tricky (.) I'm just nervous about it (.) If it works **I** can see it being very beneficial (.) so does it become a matter of policy (.) Can you override it and say "no **I** don't want to pair program on this piece of work (.) **I** want to do this by myself" (.) would that be frowned upon because maybe some people would always want to work on their own (?) **I** just want to say that I'm a bit nervous about pair programming (.) is anybody else nervous about pair programming or is it just me (?)

310

315 **S/W Engineer 100** **I** think its just you (.)

All: laugh (.)

320 **S/W Engineer 112** Looks like its just me then!

S/W Engineer 042 **I** can imagine that there would be issues (.) **I** think it can be very beneficial (.) But **I** can see that there would be situations where it mightn't work (.)

S/W Engineer 065 **we** could be like Company Y (.) have to wear smiley badges and be happy all the time (.)

325

S/W Engineer 042 That said **we** won't really know until we've actually tried it (.)

S/W Engineer 112 Most people did it in college (.) it was a pretty typical thing to do (.)

330 **S/W Engineer 065** Being the only student here (.) I've never done any pair programming in college!

S/W Engineer 100 Yes (.) but you all have a PC *each* now!

All: laugh

335 **S/W Engineer 124** **I** think some of this stuff **we** have implemented already (.) the task board (.) story cards (.) **we** know functionality that is coming into the product (.) I'm just wondering if people are happy with the stuff that's been implemented already and how do they think things are going (?)

Test Engineer 030 Well just from a test point of view (.) the only issue **we** have (.) is (.) well **we** can see how it would be easier for you but it actually makes our workload heavier (.) **we** have to come to the board (.) write out story cards (.) And if something gets rejected or whatever then it might still be in the queue and **we** have to do all this admin around the pieces of paper just for the bugs (.) So **I** don't know if there is some way that **we** could "electronicify" or put it in our Bugbase or whatever (.) So it is one area where **we** have to do more work rather than less (.) And OK (.) it might make things easier in terms of more visibility of bugs and **we** might be getting more bugs fixed (.) so in the long run it is actually helping us (.) But **we** do find that a lot of our time is shuffling things up and down in a queue (.)

340

345

350 **S/W Engineer 100** But is that not good (?) Because you are actually reprioritising things every day now (.) whereas before you only got to do it once a week (.)

Test Engineer 030 Yes (.) there are good points to it (.) but **I** do find there are a lot of administration types of work around it and it does pay out in that it allows you to be more efficient in fixing bugs (.) so there is a pay off (.) But it is causing us extra paper work which **we** aren't especially delighted about (.) Just in terms of the management of what goes into that board (.) The biggest problem is that we're working on two systems (.) **we** have a repository for bugs (.) **we** have a board where the bugs are actually going to get fixed (.)

355

360

S/W Engineer 100 Can **we** just come back to the pair programming and the task board (?) What **I** think about the task board (.) and **I** feel it myself (.) is that the engineers have a hell of a lot more autonomy now (.) In what they do (.) there is much less control about what **we** do now (.) **we** pick things off the board (.) ourselves and **we** drive them ourselves right through to the end (.) There is an element of control and management in terms of what **we** actually do is gone now which may have existed in a more traditional model (.) And **I** think what unit testing and pair programming does (.) it's a different kind of control (.) It's a control over how **we** do what **we** do (.) that **we** do it right (.) because without that (.) you can have a lot of mavericks (.) Not on purpose (.) they're not out to break the system (.) Essentially that's why they're there (.)

365

370

S/W Engineer 066 The main use of pair programming is for checks and balances (.) And its ideal for people who are less knowledgeable to pair with someone who is more knowledgeable (.) We're doing it informally (.) we're doing it from the point of view of knowledge transfer (.)

375

380 **S/W Engineer 112** But that's the end result (.) but you're not going up to somebody and saying "I have to fix a bug in this area (.) how does this work (.)" (.) and then getting the knowledge you require and then going off and doing it (.) Pair programming you'd be more (.) you both work on the problem together (.) you find it out together (.) one person presumably has a bit more knowledge in that area and that's your useful spin-off (.) that the person with less knowledge now has more knowledge (.) But you're actually working on the problem together (.) as opposed to one person working on it and then asking for help (.) There's a definite distinction there (.) As you say there is knowledge transfer as the end result (.) and that's a benefit (.) The other benefit is that it prevents us from the situation of having domain experts (.) and there is one domain expert with is a notorious creator of bottlenecks (.) Back to my nervousness (.) I'm sold on it for those reasons (.) but I'd be nervous if it was policy (.) I can actually picture myself (.) personally coming into work and thinking of the day (.) sitting down at a desk doing this (.) and I can imagine myself dreading it (.) Whereas if it was every now and then there wouldn't be a problem (.) but if it was policy (.) I can see myself (.) having a problem with it (.)

395 **S/W Engineer 113** I think it is difficult to pair program from the very beginning to the very end (.) But everybody has a blind spot (.) sometimes you are staring at the screen (.) at the blind spot and you can't see the thing (.)

400 **S/W Engineer 101** Perhaps we can make it optional (.) or only work it into part of the work of a bug fix or feature development (.) that won't make people as nervous (?)

S/W Engineer 112 Well the danger is that if its optional then no one will do it!

All: Laugh

405 **S/W Engineer 100** But that's the issue is that "is it enjoyable (.)?"

S/W Engineer 112 I can see it sometimes being enjoyable and being a very pleasurable way of working (.)

S/W Engineer 042 I think it would have to be informal (.) there may be a clash (.) you've got to keep it informal (.)

410 **S/W Engineer 100** we should use the code review via pair programming approach (.) as an experiment (.) The test case is what we get code reviewed (.) before we code up the feature or fix (.)

S/W Engineer 065 I don't understand how I can write a unit test for everything (?) I just don't understand how to do it (.) like how do I write a unit test for the build framework (.)

415 **S/W Engineer 101** You could write a test where the build script checks for the presence of a valid marker (.) for the right source directories being mounted (.)

420 **S/W Engineer 066** Put in a build marker that doesn't exist (.) I expect to have JAVAC in my compiler variable (.) we can use CPPUNIT to test the C++ libraries and JUNIT for the core JAVA classes (.) it's a critical framework (.) test instantiation and parsing of data (.) we can test that the basic functionality works (.) that stuff is fairly easy to do with CPPUNIT or JUNIT (.) we can test quickly that we haven't broken it (.)

425 **S/W Engineer 113** I just wondered about the feature cards (.) that the feature cards are presented by other people (.) all of them are very reasonable (.) after research by a meeting (.) are they not just a single person's ideas (.) Before you put it into action (.) is it coherent (?)

430 **S/W Engineer 042** The idea with these features is that we're going to sit down and thrash out (.) define the business requirement (.) the software requirement and possibly even the actual design (.) and therefore having too much detail in the actual task is going to compromise our ability to meet the real need (.) When tasks are there (.) in detail (.) you can get lazy and just do the tasks (.) whereas if the tasks aren't there (.) you go in and just see the story you immediately have an idea (.) in your own head how it might happen (.) I think it's more useful to get the story rather than the story plus the anticipated tasks (.) Its at that point that Test People can come in (.) maybe we need to define a point in requirements definition where test come in (.) maybe even right at the start (.)

440 **Test Engineer 030** Do we even want the "Spike" to combine say 3 different tasks (testing (.) explaining customer need (.) engineering the idea) (.) they're all doing this idea of requirements gathering (.) maybe even do it all on a single day (.) say a Friday afternoon (.) and complete it sufficient to commit to the actual job (.)

445 **S/W Engineer 042** Yes (.) there is still this idea of ideal days in the engineering department (.) Its very hard to use the ideal days (.) when you're used to trying to get as much work done as you can (.) It's difficult to allocate time (.) to say we're actually going to use this time to design and test the actual task (.) Even though I might have other work to do (.) I think we should take this time (.) to bring in the other parts of XP (.) Something has to happen anyway (.) you have to go to the customer to make sure what the requirement is (.)

450 **Test Engineer 030** The result of this afternoon "bartering" exercise (.) and I think that would be very useful for us to not even necessarily have any input (.) just to be present to know what is going on (.) And the result is that we need to capture some sort of artefact (.) no we're not talking about wasting days on documentation (.) I'm talking about some clarity of what is going to be done (.) something to physically have (.)

460 **S/W Engineer 042** There are supposed to be documents coming out of those meetings (.) presumably a customer requirements and a design as well (.)

465 **Test Engineer 030** we don't need bells and whistles (.) we do need clarity (.)

S/W Engineer 042 we do need the documents (.) clarity (.) not necessarily word-of-mouth (.)

Test Engineer 030 And the other thing is that if something changes in the course of development (.) that the documents contain some line noting the change (.)

470 **S/W Engineer 042** And the thing to watch out for is that some of those meetings mightn't be interesting to all concerned (.) we should be able to sit out if your not interested in the design discussion (.) The initial phase though test and Product Manager's should all be in (.) it's just during the design phase the Test People and Product Manager's might get bored (.)

S/W Engineer 112 How does the planning game work (.)

475 **Engineering Manager** You have two representatives (.) someone who speaks for the software (.) someone who speaks for the customer (.) The software person makes technical decisions and the customer person makes business decisions (.) These people meet (.) They review what is in progress (.) and they decide what new tasks to take on (.) The software person gives a certain amount of engineering capacity (.) and the customer person decides how to spend that capacity (.)

480 **S/W Engineer 112** Are we doing it (.)

S/W Engineer 125: At the last planning game we came away with enough work to keep us going to Christmas (.)

485 **S/W Engineer 112** Were there any engineers at it (.)

Product Manager 125 A manager and an engineer were there (.) The outcome of the Planning Game is the Roadmap (.) the Planning Game led by me utilises about a quarter of our capacity (.) taking up a small chunk of our time to work on key strategic developments (.) But that's not all that

490 we do through the planning game and the roadmap (.) because we reserve half our capacity for customer implementations and the final quarter for technical infrastructure or architecture (.) Engineering keep this in their back pocket because engineering is still responsible for our technical excellence (.) keeping us abreast of new technology (.) The Product

495 Managers in terms of roadmap expect a certain capacity available to the Product Managers for new strategic features (.) while allowing that we do customer implementations and technical excellence (.) In reality once it gets into engineering (.) it gets onto the board and its no longer visible as a planning game item (.) customer need (.) technical change or refactoring (.)

500 we want to deliver product (.) we are still a product company (.) but we have to balance all three needs (.) product (.) customer and architecture (.) For the next major release for example the Product Manager was told that we should plan on having X engineering resources (.) But we have to come up with a working methodology to balance the strategic need from the tactical

505 need (.) where a feature is needed for the next release of a product or a feature is just to get us in the door (.)

Engineering Manager The product manager is the product manager (.) a committee provides input but the product manager makes the strategic decisions (.)

510 **Test Engineer 030** I think what would really help us is knowing what is coming down the pipeline (.)

S/W Engineer 125: The things we publish as a product roadmap that we'll be putting in the sales pack are our strategic vision; they won't be all the on-going feature development (.)

515 -end-