

WHAT SCENES AND IMAGES DESCRIBE THE SOFTWARE DEVELOPMENT AND TEST ENVIRONMENTS ? (Firmware and VLSI at Co. X)

OUR PROCESSES HAVEN'T MATURED TO THE POINT WHERE METRICS ARE USED INTELLIGENTLY FOR IMPROVEMENT AND CONTROL

OUR FEEDBACK ON PROGRESS AND RESULTS IS OFTEN BASED ON A SINGLE METRIC - AND THE INFORMATION ISN'T ALWAYS USEFUL TO THE 'DOERS'.

People don't know where they are...they'd like to know how they're doing - not just from the point of view of 'you're late'...because we'll always hear that...it's a fact of life...(beyond that) they'd like to have a sense of what success they are having

W4-1323

Code changes..is a metric that is rolled all the way up to the executives..unfortunately I don't think it's a good metric...because...you might have code changes to [optimize some aspect of drive performance or respond to late breaking changes]..it penalizes the wrong group...the coders look bad even though it was a mechanical problem.

f32830

As for metrics -- the main thing we track is yields -[no formal metrics on the process]...Just on a reactive basis - if someone comes back from compatibility and says 'there's something wrong here' then you go fix it

m2-3150

WE ARE ENMESHED IN INFORMAL PROCESSES THAT WE DON'T ALWAYS SEE.

Something that struck me when I joined the company...the way information flowed between groups...people didn't understand that firmware was just a component of the final drive assembly...so terminology like 'stakeholder' or 'internal customer' [was foreign]...the only customer is our OEM

f20100

[if in a DFSS for software course people were told they'd be building feedback loops into their processes] they're going to get blank stares on their faces - because that's not there yet...I think what we need to do is understand what process we do have and understand that it's really not documented very well...[we're] working on it...but...not there yet

w1-fw1 1400

DFSS doesn't work, in my opinion, unless you have a very stable process to begin with"

w1-fw10149

USING THE SEI CAPABILITY MATURITY MODEL AS A BENCHMARK OUR DESIGN PROCESSES ILLUSTRATE SEVERAL OF THE MATURITY LEVELS

People don't know where they are...they'd like to know how they're doing - not just from the point of view of 'you're late'...because we'll always hear that...it's a fact of life...(beyond that) they'd like to have a sense of what success they are having

W4-1323

Code changes..is a metric that is rolled all the way up to the executives..unfortunately I don't think it's a good metric...because...you might have code changes to [optimize some aspect of drive performance or respond to late breaking changes]..it penalizes the wrong group...the coders look bad even though it was a mechanical problem.

f32830

As for metrics -- the main thing we track is yields -[no formal metrics on the process]...Just on a reactive basis - if someone comes back from compatibility and says 'there's something wrong here' then you go fix it

m2-3150

WE HAVE PROCESSES... WE BREAK PROCESSES

SOME DEVELOPMENT GAPS ARE NOT FOR LACK OF A PROCESS...THEY MAY BE DUE TO OUR FAILURE TO FOLLOW THE ONES WE HAVE

We got our first marketing requirement for 'Aspen' that is to be shipped in July...we got it yesterday (in February)...and via our design process (SLAM) it's supposed to take more than a year

f1-0600

[for detailed design, after concept review]... We don't have a good development process (for firmware)...we've tried and it's ongoing...we have developed a flow chart and used MS Project but we haven't driven that to our individual contributors...I.e. we have a process but at the moment it's not happening

f1-2000

We have two problems with the way that concepts are developed in SLAM Phase 0 -- in design concept review all of the different platform groups come in with 'here are how we believe we're going to meet the specifications...' -- but often we are either behindschedule (and rushed) or the requirements change (undoing benefits of the risk and concept analysis)

f1-1440

TO MAKE SCHEDULE WE MAY BE FORCED TO COMPROMISE ISSUES THAT WE KNOW COULD HURT US LATER

A lot of the [disruptive product development changes] come from upper management..at my level I can say no, but typically I'll get vetoed real quickly...and we end up having to march to the new beat...because the delivery date doesn't change..the only thing that changes is everything that goes with that date.

f3-1530

The end date is fixed..and the fornt end is still waffling..then when you finally get to what you have to do you may have problems in the (otherwise fairly capable) execution part of the process

f1 -0420

We have a three year plan that changes every quarter...so the resources get shuffled That's why we can't do it consistently - do it right - and then move on to the next one

th1-0837

GAPS IN OUR DESIGN REQUIREMENTS OFTEN SURFACE LATE IN THE PROCESS

IT'S DIFFICULT TO PRIORITIZE REQUIREMENTS AND OUR WORK RELATED TO THEM

There are some things that have to be changed -- but intermixed with that are things that people just think would be good ideas- it makes it kind of hectic because all of a sudden everybody's trying to do all these things - working frantically because of the quantity of things going on (some important, some less but being done anyway)

m2-1447

A lot of our designs..take..existing blocks and put them together in a new controller...with some upgrades and new features

m1-0630

We try to take an existing controller and define the changes - [sorted into mandatory and desired]

m1-1150

CUSTOMER (REQS / SCHEDULE) INFORMATION FLOW IS NOT ALWAYS TIMELY AND IS NOT BI-DIRECTIONAL (CUST <=> ENGR)

Typically the engineering teams don't get that involved with customers...when we do get involved it's more at the back end..when there's a problem we have to get engaged...but I'd say up front we don't get that involved...not as much as marketing and not as much as we probably should be....

f3-1030

[Marketing requirements come in and engineering takes a look and says..we may say 'we can't do this' but usually it ends up being some sort of negotiation..or a mandate that 'you must figure out a way to meet this specification..'

f1-0530

"Right now we don't get too involved with our actual OEM customers...we'd like to get more involved with them...right now we tend to be isolated from them because we have the marketing guys in between us and them

f3-0500

[re: latent requirements]...sometimes you just don't get enough information fed back (from OEM customers, through Marketing)...or sometimes the Marketing guys just aren't aware of the things we have to offer

f3-0710

WE MAY RELY ON CHANCE TO GET ALL THE REQUIREMENTS INFORMATION WE NEED UP FRONT

[using the feature request system to gather requirements] we're dependent on the person who entered the request to provide sufficient detail..that's not always there...we get into a schedule crunch and we get into the bottom of it and realize we don't have sufficient detail

w2-0900

We tend to generate requirements - and then invite people review them (rather than build them directly from customer inputs)

th1-1438

Requirements can come from any number of sources...the test and verification group (EDE) will give us a lot of feedback during the qualification process as to what's broken -- they don't give us a lot up front..because they don't know

w1-0710

WE PREACH TEAM PLAY... WE MAY PRACTICE "TEAMS OF 1"

WE HAVE AN 'INDIVIDUAL / SPECIALIST CULTURE THAT CAN REALLY RESIST TEAM WORK AND CHANGE.

[an example of the culture's unwillingness to expend 'energy' for 'process'...is] the Task Database...where everybody enters their tasks and start and completion times...They hate doing it...it seems to them to be an extra step...it's not obvious to the person doing the code change what use it is...

f1-3000

[per the culture's willingness to contribute to process improvement activities]...I'm not sure if it's 'unwilling' but it's not too far from that...because schedules are compressed..procedures and repeatability [are perceived to] take extra effort and time...they are things that people trying to get the drive out the door want to expend their energy on...

f1-2900

WE MAY DEPEND ON INDIVIDUALS TO CARRY THE LEARNING FORWARD FROM ONE PROJECT TO THE NEXT

Historical data becomes hysterical data when you either didn't store it..or it was stored in a way that you can't use it..or the person who knew where it was stored is gone...it becomes folklore...

f21350

[problem root-cause removal/prevention] is informal...based on the experience of the individuals involved...So if it goes into the individual's memory and they know what to watch out for next time

m2-2710

DECISION-MAKING HAS BECOME DISTRIBUTED - WHILE INFORMATION SHARING HASN'T CHANGED

WE HAVE DISTRIBUTED AUTHORITY AND INFORMATION 'WIDE AND DEEP' THROUGH THE ORGANIZATIONS -- WITH MIXED RESULTS.

The down side [of empowering people deeper in the organizations with more decision-making power] is that decisions get made locally without consideration of impacts on the total (product and people) system

w4-2113

One positive is the way that decision-making has been driven deeper into the organizations...involving more people...and opening up gateways where people can impact a product

w4-2049

INFORMATION TRANSFER BETWEEN DEPARTMENTS OR FUNCTIONS CAN BE SURPRISINGLY CONSTRAINED.

[a gap between test and FA is...if we detect a materials problem...I may have someone with enough insight to say "I know what's going on here"...that information has no place to go For example..media defects after the process...we may have seen the problem before...[but with no way to communicate that to FA]...they may spend weeks trying to figure it out

m2-2210

I'd say that there is no formal communication procedure or line between departments [for resolving/removing problems or preventing future occurrence] For example if you discover something about how they are doing - and you might have a solution for it - but there is no formal way or good way to get that information to design engineering in a way that it would be picked up and utilized.

m2-1600

To insulate against decisions and dependencies in other groups - we may put up polite but adversarial walls

w4- 3020

WE DON'T CAPTURE OUR ENGINEERING INFORMATION CONSISTENTLY OR IN A MANNER THAT SUPPORTS SHARING

[Another example is] documentation -- I've got 4 or 5 documents that I'd like to be able to sit down and write to I just don't have the time. And you keep [moving] ahead and you pay for it at some other point

th1-2450

The difference is..the document is not well written..it takes 2 or 3 times to figure it out..and many times the wrong interpretation..in a concise document the reader understands it in a short time...and it's reusable [today] you read it once and you don't want to go back.

th1-1750

[It's done by] looking at specs for existing controllers -- we don't have a good level of documentation and that's one of the things [we are working on] - better documentation at the block level.

m1-0721

WE ARE BEGINNING TO APPRECIATE THE VALUE OF SYSTEMATIC DEVELOPMENT METHODS

After a program is complete we do a post mortem on it..it's feedback we use to try and get things better on the next one...it is an area we are working in...it's probably not wired back very well right now...one big hole is we [do post mortem from the product side and present to platform directors] but don't do one from the platform side

f3-3040

We are making some progress with discipline in requirements development[using an SRS (based on IEEE 830)and a design document (based on IEEE1610)...people are becoming more familiar with the terminology and with the payback for doing this

w1-0450

once you have data...per inspections] then you can go off and look at - do you have a systemic problem or do we have a tool problem? If you could achieve that...Then perhaps [the statistical tools] would play a bigger role

w1-2420

We're just starting to use MS Project..to tie all the functional silos together so that all the individual projects are visible to the design center..we're pushing down the responsibility of inputting data down to the engineer responsible for that project...

f31940

CHANGE IS REALITY...

AND OUR PROCESS DOESN'T SUPPORT IT

PDP DOES NOT SUPPORT CRITICAL VISIBILITY TO SW/COMPONENT ISSUES

Configuration management problems at the customer... "if you don't know what to ask, you don't get the answer you need." (It was not clear that we knew when we were testing with the correct configurations)

The customer has a clear qualification procedure, but it relies on testing with their backplane.

(Specification is not independent of the customer implementation)

COMPONENT VERIFICATION SOMETIMES BECOMES AN ISSUE LATE IN THE DEVELOPMENT CYCLE

Core team members don't have time for asic verification "People are on other things."

Resources for chip verification were limited until the last few months... We now have a firmware test team.

In the past when a new chip arrived we would stick it on a board and say "now what do we do with it?"