

Going Agile at F-Secure

Maaret Pyhäjärvi, Senior Quality Engineer, Project Office

<maaret.pyhajarvi@f-secure.com>



Outline



- Introducing agile product line development in practice
- Avoiding fragile on route to agile - lessons learned
- Challenges tackled and ahead in in-team and cross-team work

Introducing Agile Product Line Development in Practice



Context of the story:

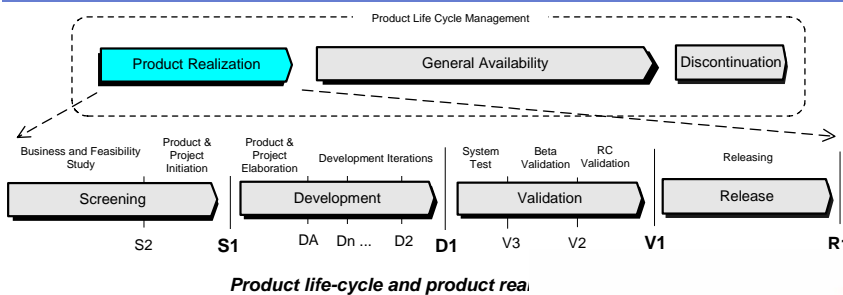
- WinCore product line delivers Windows Workstation and Server Security products in various flavors
- Products are part of systems, that in addition to these products include backend systems, management systems and integration to company services
- Various markets for various products from the product line: consumer, corporate, ISP
- International multi-language product, supporting various operating systems and required to work with various kinds of software out there in the wild



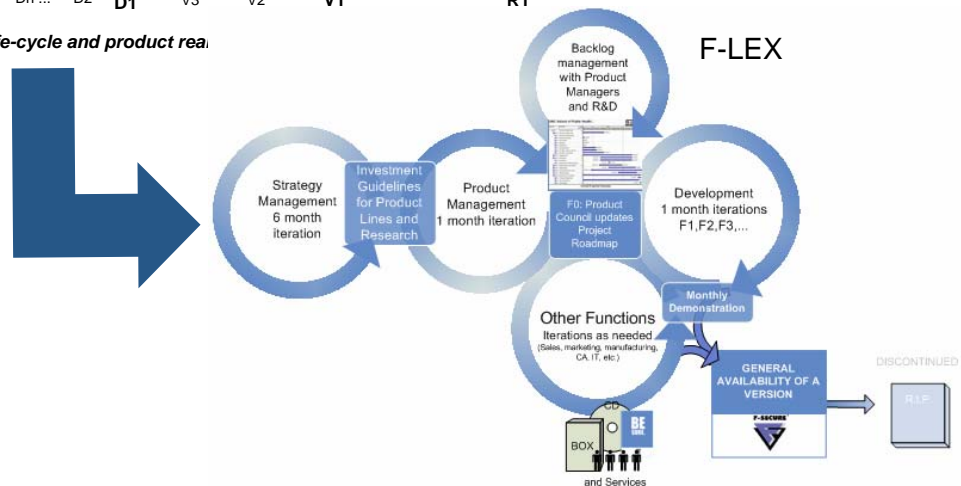
How we got where we are now:

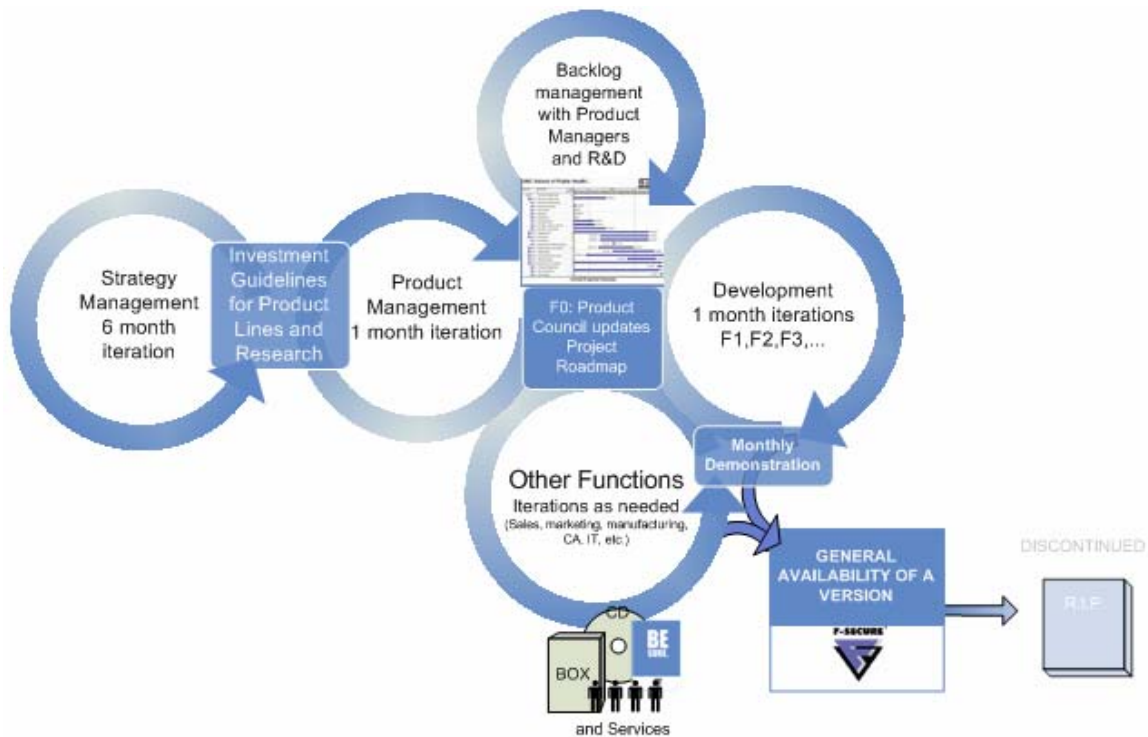
- Agile piloting in some projects in 2005
- FSC new process F-lex launched beginning of 2006
- WinCore went agile, and simultaneously product line one year ago
- “Nuclear explosion in heart of R&D, business is still shaking” – we just did it.

FSC development process change



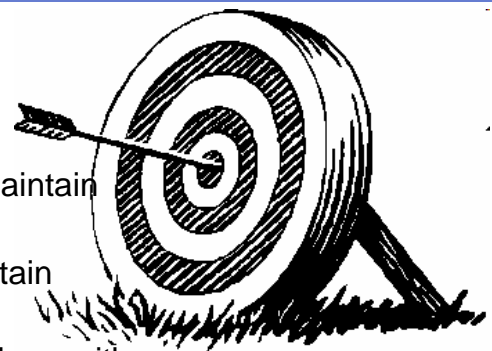
FPRP – F-Secure Product Realization Process



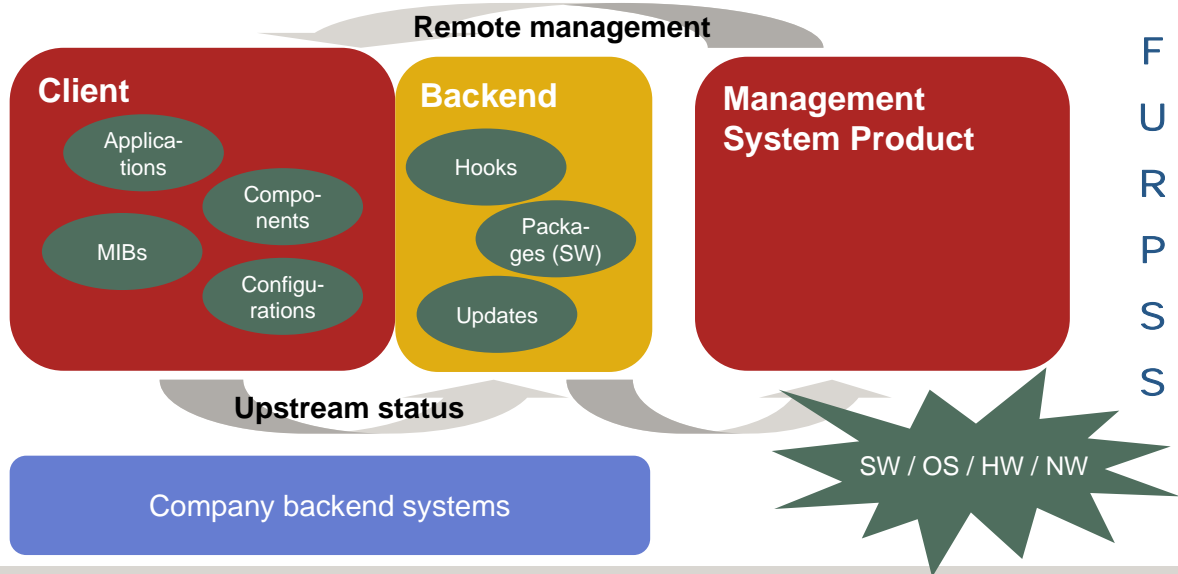


Key activities and concepts that we are focusing on

1. **Incremental** development. Ready for at least internal release after each sprint.
2. Helping Product Management to make and maintain good **Product Roadmaps**
3. Helping Product Managers to make and maintain excellent **Product Backlogs**
4. Planning **project roadmaps** and discussing them with Product Managers.
5. Concentrating on doing the **retrospectives** and **planning days** very well.
6. **Visibility** should be a major agile principle. Really a problem at the moment.
7. **Backlogs** for common components
8. Defining more exactly what **DONE** means for a Product Backlog item.

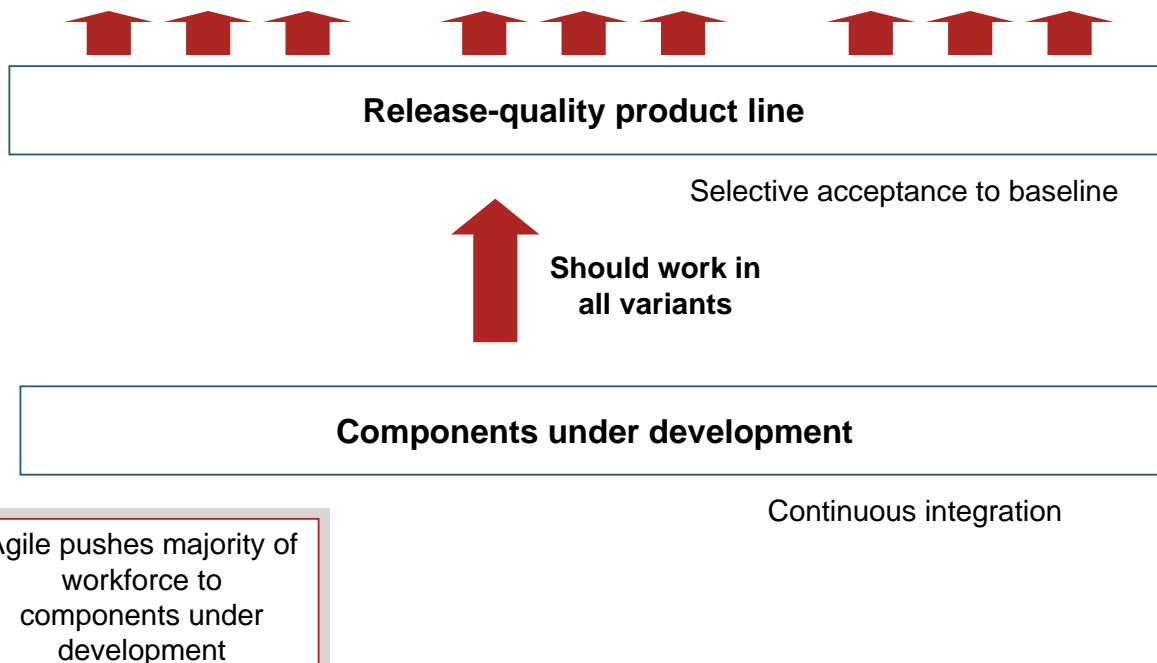


What "System Testing" Means to Us?



System testing means considering the overall experience the customer will get from all product components in their environment of use
To test a system, you need to break it down to parts, but when preparing, you need to consider implications for other parts in addition to the one you're considering now.

Agile and Product Line Development



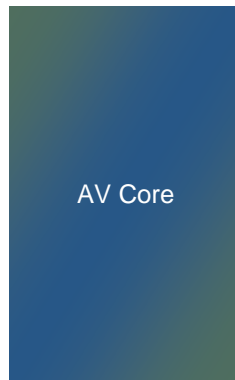
AVRM Quality Debt

Released products

- AVCS 6.0
- PEX 6.0
- PEX 6.20
- IS2006 (PEX 6.10)

- Environment changes
- Configuration modules

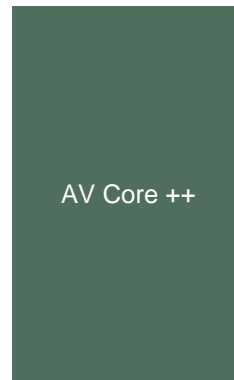
AVRM result



Old branch maintenance releases

- AVCS 6.01
- IS2006 SP1 (6.12)
- IS2006 SP1 for SP (6.12b)
- AVCS for SP 1.0 (6.11)

Where we should be



Core releases

- AVCS 6.01
- AVCS for SP 1.0
- IS 6.12
- IS for SP 6.22

Goal: Core releases start quality is equal to previous releases' release quality



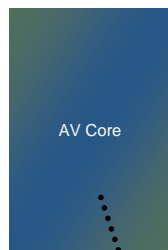
What seemed to be happening

Released products

- AVCS 6.0
- PEX 6.0
- PEX 6.20
- IS2006 (PEX 6.10)

- Environment changes
- Configuration modules

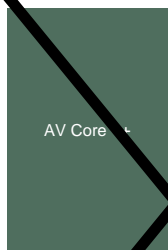
AVRM result



Old branch maintenance releases

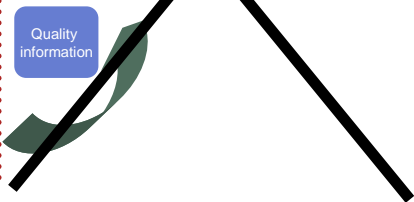
- AVCS 6.01
- IS2006 SP1 (6.12)
- IS2006 SP1 for SP (6.12b)
- AVCS for SP 1.0 (6.11)

Where we should be



Core releases

- AVCS 6.01
- AVCS for SP 1.0
- IS 6.12
- IS for SP 6.22



- EPM features
- AV changes
- HIPS
- BL
- New installer
- ...

Core feature creation projects – all start to build on unknown quality and all start at same time

Goal: Minimal amount of tracks at once!!
AVRM quality debt needs to be addressed!

Requirements / Project Plans Misconceptions with Agile



Product backlog vs. old-approach

- Product backlog actually replaces project plan, not requirements specification
- Defines “this we want to deliver for the user”, not “this is what the user wants to do”

Product Owner vs. old-approach

- Product owner role is held by Product Manager (priorities) and Project Manager (delivery organizing, dependencies)

Our Product Backlog – two tiers



PdM-PBL

Delivered value
for customers

*Synchronize delivery
for customers*

*Features “this we can sell”
Debt “this is needed for getting
something to sell” – timing-wise we pay
interest
Release cost “Making a release from
core costs 2 mm of testing”
Basis for release roadmapping*

Area-PBL

Area 1



Area 2



Area 3



Area 4



CoolFeature1

Release A

Release B

Quality debt

Legend

Avoiding Fragile on Route to Agile - Lessons Learned



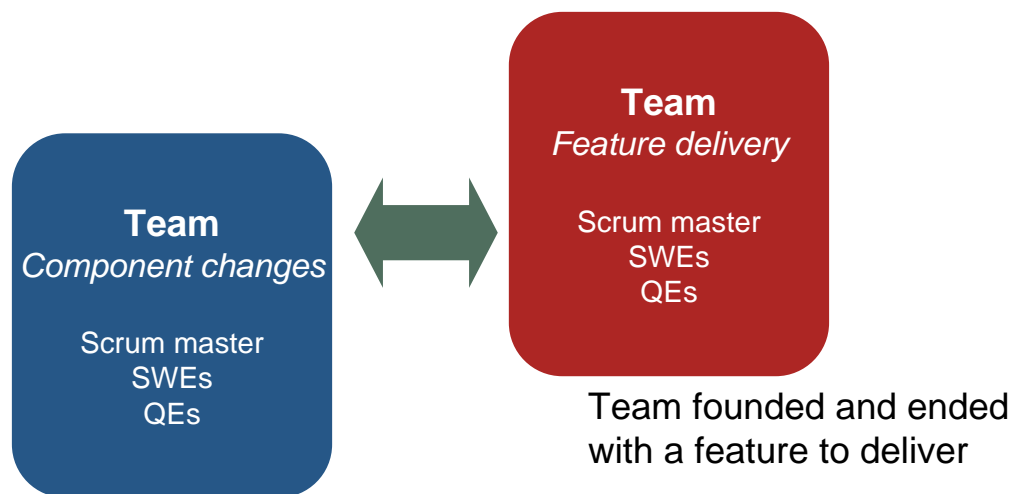
PITFALLS TO AVOID – OR CLIMB UP FAST

- Underdoing or overdoing documentation, not focusing on value for the context at hand; differences also within organization
- When avoiding “requirements” for some reason, coming up with “constraints” and other terms
- Not doing architecture planning before implementation
- Not having a vision for the product and shared goal for a sprint
- Several product backlogs with conflicting requirements make priorities unclear
- Team chooses what they will do, and assume someone knows to take the rest to make a whole; assuming a team that gathers what was left
- Team not delivering feature for the whole system, just components
- Everyone writes their documents in place they choose, no common locations
- Must-have inefficient daily meetings just because daily meetings are Scrum
- Giving up on functional specialties all in all: cross-functional teams where everyone can do everything does not grow specialists but generalists
- Starting too many goals at once for being afraid of close deadline
- Not planning ahead, just one sprint at a time
- 80 % done goal and then postponing rest of it for 3 months, since it is not schedule critical any more
- Thinking of other teams as “chickens” and working together only on planning day
- Not allowing fixing wrong or stupid sprint goals after sprint has started
- Not allowing to think and talk of future
- Leaving wrong work for future because of poor understanding of sprint goal
- Faked self-organizing: planning days are task order dispatching sessions
- Relying on improved quality on plans for end-of-project testing and not accounting for added changes throughout the project
- Unclear resourcing practices, cross-team resourcing, same person expected in several teams
- Weekend sprints as extension to resourcing

Overdoing and misunderstanding results in fragile, not agile

13

Challenges Tackled and Ahead in In-team and Cross-team Work



Majority of team fixed, e.g. Virus Protection team, WASP team

Tried teams of 2 (too small) to 22 (too large) members. Optimal less than 10, minimum 4.

Challenges Tackled and Ahead in In-team Work



Tackled

- Making teams work as teams, not collection of individuals
- Having system focus for area to deliver, taking responsibility
- Delivering components for baseline without breaking the system (for most parts)
- Software engineer / Quality engineer collaboration, sharing information within team for one goal
- Team rooms without quiet space or time
- Quick private builds for the whole product
- Contact with customers and relevant stakeholders, continuous deliveries

Ahead

- Delivering cross-product line components on a fast pace without breaking our systems
- Facilitating everyone's opinions and commitment
- Personal career development; agile makes everyone "developer", career paths require focus
- Right drive for customer involvement for fast releases
- Enabling innovation with continuous short deliveries

Thursday, 30 November 2006 Page 15

Challenges Tackled and Ahead in Cross-team Work



Tackled

- Everyone not comfortable with variable teams, they rather do cross-team coordination that cannot be fully avoided with team compositions
- Teams talk directly to each other on delivery items, not assuming coordination through PjM nor PdM
- Too many project managers
- Management retrospectives in addition to team retrospectives
- Cross-team information sharing about product structures, training what are changes for those outside the team; lot of implicit information to transfer on a regular basis
- Daily integration product builds for the whole system

Ahead

- Division of planning, coordination and facilitation work between PjM / CQE / SM roles
- Easing up the complexities in long-term planning of multiple releases of multiple products from product line
- Right contents on SoS (Scrum of Scrums) coordination meetings on a daily basis
- Mechanisms of finding areas outside teams expectations naturally: QE weekly works, need the other perspectives

Thursday, 30 November 2006 Page 16

Introducing agile product line development in practice

- Big changes → be prepared to fix problems fast and use creativity on all levels!
- No quick route to multiple releases from product line with multiple teams – it's just communication

Avoiding fragile on route to agile - lessons learned

- Product backlog work is essential – appearance may fool you
- Cultural change toward agile different on different levels of organization, none can escape
- There's only so much you can shorten the duration of overall delivery, benefits are in incremental delivery adding trust

Challenges tackled and ahead in in-team and cross-team work

- Quite different in small-scale and large-scale
- Challenging, rewarding and productive!