

SPLC 2006



Agile and Software Product Line Methods: Are They So Different?

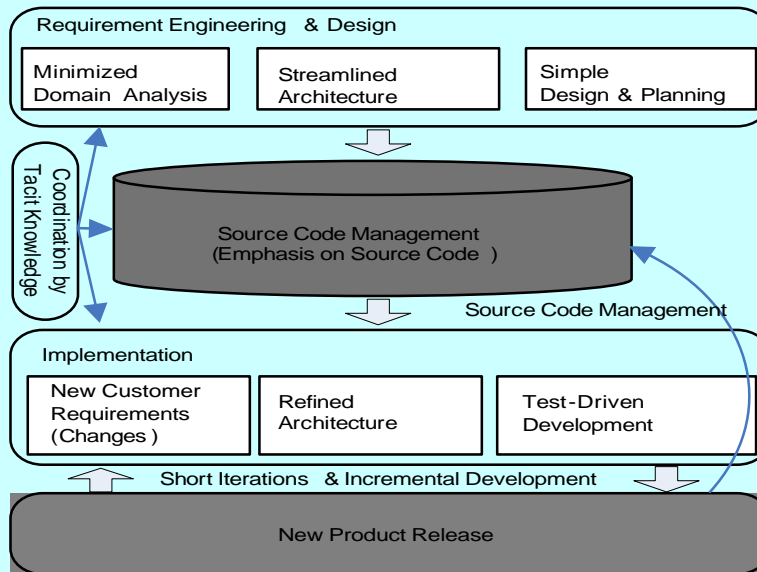
Kun Tian and Kendra Cooper
The University of Texas at Dallas

**1st International Workshop on
Agile Product Line Engineering
August 21, 2006 Baltimore, MD**

Problem?

- **Need to rapidly deliver high quality software that meets the changing needs of stakeholders in a specific domain**
 - Adapt to changing requirements due to changes in business environments, new/revised competing products, etc.
- **Numerous proposed solutions include Agile and Software Product Line approaches**
- **Ideally, an approach that integrates agile and product line principles would be useful**
 - Here, we take the first step: compare the two approaches from engineering, software quality assurance, and project management perspectives**

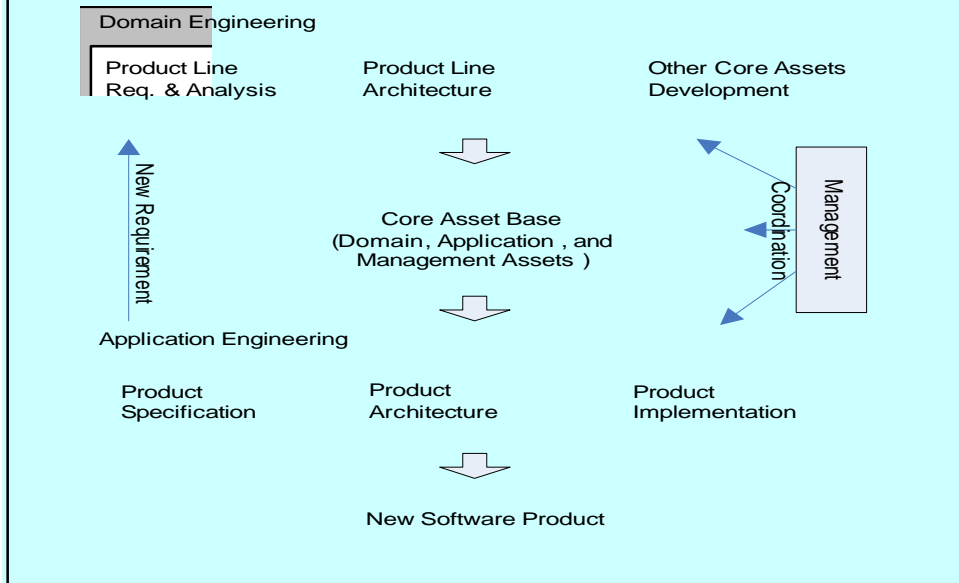
Agile Development?



Agile Development?

- Focused on delivering **working code**
 - Advocate minimal process definition, modeling, and documentation
 - Adaptive rather than predictive methods based on 12 principles
 - Satisfy the customer through early and continuous delivery of software
 - Short iterations
 - People over processes and tools
 - ...
- Developed to adapt and thrive on frequent changes/churning
- People-oriented rather than process or tool oriented
 - rely on people's competency, expertise and direct collaboration to produce high-quality software
- Numerous Agile Methods have been proposed
 - Extreme Programming, Agile Modeling, SCRUM, Crystal Methodologies, Dynamic System Development Method, Adaptive Software Development, etc.

Product Line Engineering?



Product Line Engineering?

- Support developing a set of systems that share a common, managed set of features
 - Systems satisfy the needs of a particular market segment or mission
 - Systems are developed from a common set of core assets
 - A core asset is a software artifact that is re-used in the production of customized products in a software product line (SPL).
 - Core assets include the requirements, architecture, components, modeling and analysis, plans, etc.
- Offer improvements in time to market, cost, productivity, quality, etc.
 - Rapid market entry, flexible response via mass customization
- 3 major activities: core asset development (domain engineering), application development, management

Comparison – Engineering

	Agile Methods	Software Product Line Approaches
Software Requirements	Emphasis on quick response to requirement changes with short iterations and small increments for the application. Direct customer collaboration; customers must participate in the whole software project lifecycle.	Application requirements and domain requirements are both engineered. Indirect customer collaboration using well-trained customer representatives.
Software Architecture	Minimal emphasis on the application architecture features beyond the immediate iteration.	Application architecture and domain architecture are both engineered.
Implementation	Iterative and incremental implementation; minimum effort on plans and models, using “just enough” approach.	Plan-driven implementation with an emphasis on application engineering process (application designs, architecture, etc.) developed using core assets. Feedback from the product-specific requirements is used to update core assets; substantial effort on plans and models.
Reuse	Optimistic use of COTS; applies streamlined domain engineering activity, without emphasis on development of reusable core assets.	Special emphasis on maturity and reliability of COTS; foundation of the method is re-using core assets defined in domain engineering activity.

Comparison – Software Quality Assurance

	Agile Methods	Software Product Line Approaches
Configuration Management	Minimized and streamlined CM approaches to do core CM tasks.	Extended CM approaches primarily for managing core assets.
Asset (Artifacts) Management	All are managed by the team as a whole; everybody could have access to them. It relies on tacit knowledge in the team to be conscious of the status of the artifacts.	Artifacts are identified as core assets, which are formally managed and kept in the core asset base. Each core asset has an attached process to demonstrate how it will be used in actual product development.
Testing	Unplanned activities; tests are carried out frequently and primarily during implementation phase to ensure that the code works.	Decentralized and planned activities; tests are reusable.

Comparison – Project Management

	Agile Methods	Software Product Line Approaches
Technology and Tools	Emphasis on simple and efficient tools. Technology impact only affects current project.	Emphasis on robust and powerful tools that support parallel works. Technology impact expands as more products of a SPL are turned out.
Project Characteristics	Not suitable for large, distributed, security or safety critical projects.	Not suitable for small, single product oriented projects.
Development Team Organization	Agile developers are good at using tacit knowledge, they share the responsibilities to the outside environment as a whole.	SPL developers follow plans and processes well, each of them has a set of specified roles and responsibilities.

Conclusions

- Tailoring doesn't need to be an "all or nothing" approach
 - Investigate three main activities in SPLMs for tailoring/introduce agility
- For example, core assets in ASPLM could be simpler
 - represent the most significant commonalities in a domain, rather than a comprehensive set (e.g., a 80/20 heuristic)
 - leave room for further development work to meet customers changing requirements, rather than pure customization
- Seems possible to tailor/introduce agility into a SPLM approach, define an Agile Software Product Line Methods (ASPLM)
- ASPLMs could be used on a wider variety of projects, such as safety or security critical systems, larger-scale, or with distributed development teams
 - For example, could apply ASPLM to financial or medical product lines

Future Work

- Define and validate an ASPLM that includes agile principles
 - All three main activities of the SPL approaches need to be investigated for possible agility: domain engineering, product development and management
 - Validate using a financial application, e.g., credit card reconciliation
- Two possible approaches
 - Take an existing SPLM process (e.g., SEI) and introduce agility
 - Take an agile process and tailor it for software product lines
- Define a set of guidelines to assist with tailoring the ASPLM

SPLC 2006



Contact Information

Kun Tian
The University of Texas at Dallas
skearth@gmail.com

Kendra Cooper
The University of Texas at Dallas
kcooper@utdallas.edu
www.utdallas.edu/~kcooper

Thank you!