Abstract

This case study presents transition of a multi location (India and Germany) project team from V-model to Lean. It highlights the improvements in quality, cost and time to market that were achieved as a result of this transition. Significant improvements done in planning, project structure, knowledge management and time to market are described in the paper. Challenges faced and challenges ahead, lessons learnt and success stories are also included.

Author: Uma Viswanath (uma.v@siemens.com)
Transition: End to end Lean implementation

V-model was used as the process R&D for almost 15 years. A need to achieve quicker time to market prompted the team to change the process model. Since the team focuses on complete product lifecycle, Lean was the perfect choice.

The situation which needed the change:

The teething problems which large projects mainly faced were:

- Experience of content changes: Due to long duration of releases (12-15 months) contents would change during the course of the timeline.
- Wish list of content and timeline: Based on market requirements, additional requirements would be added in the course of the project.
- Unrealistic workload distribution: Almost 40% of the project time was spent on planning such long releases.
- Defects caused by too late integration: Lot of features kept parallel for long durations (6 months) were integrated before start of system test, leading to too many defects.

It was time to move out of the vicious circle.

The process definition:

The Lean Product Lifecycle was defined in co-operation with the process and management experts. The lean guiding principles became the base of the definition. The 9 core elements are the guiding principles of the lifecycle, which addresses Product (Efficiency), Process (Effectiveness) and People (Capabilities).

Major changes:

- Transformation in the life cycle: Adoption to iterative development
  
  The goal of iterative development is to fix and shorten release cycles and to maximize productive development time. The approach brought changes in 2 dimensions:
  - Change in release duration, to quicken time to market: The release cycles are set to 6 months, as compared to 12-15 months in the earlier lifecycle.
  - Adaption shorter cycles of analyze→design→code→test→release: A takt 1 for iterative execution is adopted.

- Transformation in planning: Adoption to Just In Time planning
  
  The concept of requirement ranking and keeping scope that can be accommodated in the 6 months release was the new major change. The approach adapted is release

With the value driven approach, the backlog structure is defined. At each level there is a clear and singular responsibility for backlogs.

- Product backlog (PBL): Contains all and only requirements from the market, prioritized based on business needs.
- Release backlog (RBL): Is a subset of the PBL which contains market requirements as well as improvement aspects like defect fixes, refactoring and architecture improvements.
- Technical release backlog (TRBL): In many cases, the requirements are long and cannot be fulfilled in the takt duration of 4 weeks. To give the teams shorter goals to achieve and enable continuous integration, the requirements are broken down to user stories which can be realized in 2-3 weeks.

1 A takt is a time boxed development cycle of short duration, usually 2-4 weeks only. Takt is similar to sprint in agile.

Figure 1: 9 Core elements of Lean

Figure 2: Iterative development

Figure 3: Shift to Value Driven planning
Team backlogs (TBL): Contains task breakdown of the user stories.

Transformation is team structure: Imbibe ownership and responsibility
The competency driven team structure is replaced with cross functional teams who can execute a complete requirement among the team. The takt team is the smallest in-house company empowered to make decisions.

The rough phase: Challenges faced
Handling technical debt: The products are more than a decade long and with brings with it legacy defects and problems in code quality. The following measures are set up to counter the debt incurred:

- Effort allocation in the RBL for product improvement like defect fixes, refactoring, test automation and architecture checks
- Introduction of relevant metrics like Code Quality Index2 to measure and monitor code quality
- Multi level testing: Testing from developer and testing before integration ensures the quality of the merge. Further regression test on the product helps to access the stability damages caused by the merge. These tests are done during development takt. The hardening test (system test) ensures complete testing of the product during which no requirement realization is done.

- Long running tests to detect legacy defects

Achieving Just In Time planning: The intention of iterative planning is to avoid waste due to planning way ahead in time. However this was a major challenge, since the experts who do the planning are fire fighting with issues of technical debt. Handover of the user story to the takt team on time was also a major challenge. Due to this the team would wait for the requirements and also effort was lost waiting for clarifications during the takt. The measure taken to make planning JIT are:

- Planning cycle for the next release began well in advance: When it was realized that starting the planning for next takt 1 month in advance is not sufficient, planning began 2 months in advance
- Earlier interaction with the lifecycle managers for requirement clarification helped to get a better view on what needs to be planned and realized.
- Smaller requirements and user stories also sped up the planning process

Knowledge management: With the change in team structure, the expectation is that the team can achieve the requirement completely by themselves. Team expertise plays a major role in lean, since the team plans and executes and is empowered to take decisions. This is also the basis of rotation of roles within the team. This was a great challenge in domain intensive product. To address these problems, following measures were set up:

- Intensive know how transfer are planned as part of backlog when a new component is taken over.
- Webinars: Mostly of duration 1-2hours these helped teams get an overview of several components
- Code doctor and usability doctor: This initiative driven by the architects and usability expert focus on poor code/user screen and give tips to team on how it can be healed
- Creation of component know-how documents by the component owners further helped if anyone in the team wants to refer.

Communication and collaboration: The role of a product owner is extremely vital in the lean environment. He is the first point of contact for all clarifications. The roles existed in Germany only. To solve this issue, the concept of Part Product Owner (PPO) was introduced. The PPO is local and acts as a communication channel between the takt team and the PO. This has been one of the best working initiatives.

---

2 CQI is a suite of static code analyzers which gives a number to indicate the health of the code
The sweet taste of success:
Generating customer value without waste:

Before:
- Long release cycles and problem resolution cycles
- Long time for final agreement on scope for big releases
- Waste due to cutting of scope that cannot be implemented/finalized in release
- Low transparency on realized customer value during development phase
- Majority of defects found late in system test phase

Now:
- Lean SW dev. Model based on 9 core elements
- Shortened TTM due to faster planning phase and iterative model
- Ranked requirements for highest achievable customer value
- High transparency in development phase

Improve efficiency: “Improve Time to Market”

Before:
- Long time for final agreement on scope for big releases
- Big batches of requirements are processed
- The phases take several months i.e. several months between detailed planning and implementation and between development and defect fixing

Now:
- Faster agreement on scope due to clear QCD responsibilities
- Shortened TTM due to shorter planning phase (rough planning) and iterative model
- Small batches (4 weeks) reduce waste (no working on hold)
- The parallelization of planning allows more productive takts per release

Improve efficiency: “Effort for release planning”

Before:
- For all requirements, we have to create a development specification
  → requires much time for clarification, review, rework, estimation and cancel discussions

Example: 448 requirements discussed, cut down to 250 requirements → -56%

Now: We have to describe details only for requirements we have to implement (first estimation)

Example:
- 24 requirements defined
- 22 requirements implemented

Improve capabilities “Empowerment / Communication”

Before:
- Team members are spread over several locations and groups
- Dependency on experts
- High synchronization and communication effort
- Typically a topic will be discussed between different teams (2h - xd)
Now:
- People working on one feature are sitting together as cross-functional team in one location
- Everybody has a role within the team
- Team plans and decides/commits on what can be realized within the next takt
- Daily stand-up meetings for fast status exchange
- Improved communication

Example: Typically a topic will be discussed directly within one team (1h)

Conclusion:
Lean is 10% process and 90% mindset. The whole system is driven by empowered team with decision making ability and ownership. It is always said, “Lean is a journey, not a destination”. The main focus is reducing waste and increasing/generating value to customer.
As long as the focus remains in the direction of continuous improvement, we live lean!
Abbreviations

CQI  Code Quality Index
JIT  Just In Time
PBL  Product BackLog
PO   Product Owner
PPO  Part Product Owner
QCD  Quality Cost Delivery
RBL  Release BackLog
R&D  Research and Development
TBL  Technical BackLog
TRBL Technical Release Backlog
TTM  Time To Market