



Perspectives on Productivity and Delays in Large-Scale Agile Projects

Deepika Badampudi, Samuel A. Fricker, Ana M.
Moreno | June 6, 2013 | XP 2013 | sfr@bth.se

www.bth.se

BLEKINGE INSTITUTE OF TECHNOLOGY

About us



- Deepika Badampudi
PhD student at
Blekinge Institute of Technology



- Samuel A. Fricker
Assistant Professor at
Blekinge Institute of Technology



- Ana M. Moreno
Professor at
Universidad Politécnica de Madrid



- Anonymous industry partner
international, multi-industry
software and embedded systems

Agile Methods

- Small, competent, and collocated teams
- Well-collaborating customer
- Rapid value creation by prioritizing business value and early releases

- Agile methods
 - Lightweight, fast, and nimble development
 - Emerging shared understanding through continuous feedback
 - Flexibility by underspecification (avoiding requirements to be specified specified up-front)

Method Misalignment

- Our anonymous industry partner
 - Large-scale, distributed collaboration, requiring coordination among teams
 - Competence gaps due to product novelty and new technology (both for customer and staff)
 - Project customer rarely on-site
 - Many stakeholders to be satisfied in addition to project customer

- Method misalignment affects project performance negatively or leads to failure

Productivity Problems and Delays

- Bad requirements engineering, prototyping, and reuse
 - increase need for error correction and rework
 - disable effort estimation, negotiation, progress tracking, and quality assurance
- Fluctuation, unrealistic plans, and late problem resolution
 - Aggravate delays
- Inadequate software architecture
 - Limits the number of developers
- Tools are ambivalent

- Productivity problems and delays impact
 - Quality of investment decisions
 - Product release and quality
 - Market share and revenue

Performance Improvement

- Mature areas: prescriptive frameworks
 - CMMI for benchmarking-based improvement
 - "Best Practices" like
 - Leffingwell's framework for scaling sw agility
 - Sutherland's approach for offshore hyperproductivity

- Immature areas: inductive improvement
 - Learning organizations approach
 - Collect experience
 - Modify behavior to reflect gained insights
 - Basis for establishing prescriptive frameworks

Research Approach

- Goal: understand productivity impediments of Scrum projects in large-scale software product development
- Context: inductive process improvement of anonymous industry partner
- Approach: Grounded Theory
 - Perspective: practitioners
 - Phenomenon: productivity problems and delays
- Results:
 - Support specific process improvement
 - Empirical basis for situational framework for flexible large-scale development

Research Outline

- RQ1: Which challenges led to productivity problems and delays?
- RQ2: How were the involved project roles affected by these challenges?

- Iterative data collection and analysis
 - Semi-structured recorded and transcribed interviews
 - Questionnaire refined based on analysis results and information needs
- Model of causes and effects
 - Open coding: challenges, causes, impacts
 - Axial coding: connection of cause-effect chain
 - Selective coding: identification of central traits

Threats to Validity

- Reactivity: researcher presence affects subjects
 - Stay at case organization for prolonged period of time
- Respondent bias: subjects guess "right" answers
 - Study goals aligned with participant interests
 - Triangulation across subjects
 - Verification by participating in project meetings
- Researcher bias: preconceptions and assumptions
 - Company's quality manager selected interviewees
 - Open ended interview questions
- Reliability: carefulness and honesty of research
 - Chain of evidence in managed research repository
 - Checking of results by industry partner
- Generalizability: validity beyond studied case
 - Multiple projects for different industries included in research
 - Future research should replicate study across companies

Case Organization

- Company
 - Global 500 company with leading technologies
 - Diversified portfolio of software products, systems, and services built for various markets
- Products
 - Established for up to 5-10 years
 - Software and hardware, up to 5MLOC
 - Critical parts of larger systems
- Projects
 - Requested by product managers
 - 25-100 members in up to 10 Scrum teams
 - Distributed with up to 4 development sites

Case Organization

- Global project teams
 - Product manager: responsible for product success
 - Project manager: scrum team coordination
 - Architect: overall product/solution design
 - Integration manager: composition of solution
 - Technology manager: responsible for development organization
 - Independent organization: compliance to regulations
- Scrum teams
 - Product owner: product manager
 - Scrum master: development process
 - Developers: design and implementation
 - Testers: quality assurance

Results: Selected Challenges

- Requirements stability
When there is a change, it takes a couple of sprints to align everything together. The impact of change can be felt for a longer time.
- Decision making
All the key stakeholders from all teams should be involved in release planning along with the product manager and customer. That would lead to a concrete plan.
- Clarity of done
People don't want to report yellow or red. If you have many teams that all report green, but still have open tasks, this does not give a correct indication of work done.
- Requirements selection
How to split the requirements, how to phase them across different phases of the project. I would say continues to be a challenge.

Results: Challenges

- MDRE, GSE, and innovation challenges remained with the agile development approach
 - MDRE: requirements quality, non-functional requirements, estimates, requirements stability
 - GSE: communication quality, decision-making, team dynamics, progress measurement, documentation quality
 - Innovation: domain and technology knowledge
- New challenges
 - Project management: test infrastructure sharing, team stability
 - Product management: competitors' influence, clarity of done
 - Iterations: requirements selection, testing completeness, integration

Results: Selected Causes

- Project complexity → requirements stability
Requirements changes might also come from the development teams. Sometimes a team realized that they needed support from other teams or other components.
- Multiple sites → decision making
Sometimes meetings are not done jointly due to time differences. Then only the minutes of meeting are shared after the meeting.
- Multiple teams → clarity of done
People don't want to report yellow or red. If you have many teams that all report green, but still have open tasks, this does not give a correct indication of work done.
- Product characteristics → requirements selection
How to split the workflows into requirements, how to phase them across different phases of the project continues to be a challenge.

Results: Causes for Challenges

- Project complexity
 - Requirements quality, requirements stability
- Multiple teams
 - Integration, clarity of done, test infrastructure, progress measurement
- Multiple sites
 - Communication quality, decision-making, team dynamics
- Product characteristics
 - Non-functional requirements, documentation quality, requirements selection, testing completeness
- Knowledge limitations
 - Domain and technology knowledge, estimates, integration

Impact of Challenges

- Need for predictability, dependability, stability, and effective use of an appropriate amount of resources.
- Impacts differed for local Scrum teams and global teams

Scrum Team: Examples

- Requirements stability → developer
Planning: deviations from software design and project schedule
The reasons for deviation are evolving requirements and some technical challenges
- Decision making → developer
Planning: project plan was not concrete enough
Involvement of all the key stakeholders from all teams ...will lead to development of a concrete plan
- Decision-making → developer
Shared understanding and coordination: team coordination and component consistency problems
Workshops should be conducted by having all stakeholders in one place
- Clarity of done → developer
Software quality assurance: failed acceptance of features
Perspectives of developers and product managers differ sometimes. What we consider done is not by a product manager.

Impact on Scrum Teams

- Shared understanding
 - Unclear and unstable requirements
 - Insufficient qualifications and competence
 - Lacking participation of decision-makers
- Plan stability and adherence to plans
 - Uncertainty of estimates
 - Inadequate time budget
 - Deviations from project schedule
- Quality assurance and uneven work distribution
 - Splitting requirements implementation over multiple releases

Global Team: Examples

- Requirements stability → architect
Planning and coordination: solution redesigning during development, replanning, increased coordination effort
Changes in NFRs caused refactoring of design and code
- Requirements stability → technology manager
Capacity: congested backlogs
The impact of change can be felt for a long time
- Clarity of done → technology manager
Coordination: wrong understanding of progress
If all [teams] report green but they still have some open tasks then at the end to the management it is all green.
- Competitor's influence → product manager
Planning: changes in time-to-market and priorities.
The competitors come up with a similar product. Then the product needs to release earlier with at least the same features.

Impact on Global Teams

- Product design and project plans
 - Market changes and resource problems
 - Failed external regulatory tests
 - Learning process and ripple effects of changes
- Scrum team enablement and coordination
 - Sub-optimized plans and inconsistent reporting caused misaligned work, inconsistent work results, and wrong understanding of real progress
 - Insufficient communication introduced problems of shared understanding between central mgmt and remote teams

Discussion

- Many of the reported challenges were well known
 - MDRE, GSE, Innovation
 - Agile added: product management angle, stepwise implementation of complex requirements
- Development organization desired predictability, dependability, stability, and effective use of resources
 - Chosen solutions: planning, coordination, and communication
 - Cause for troubles: complexity of product and organization

Discussion

- Many determinants for productivity and delays were similar to those of pre-agile projects
 - Unclear product and product use led to requirements change
 - Unclear requirements, limited domain, technology, and organization knowledge
 - Uncertain estimates, unstable plans and integration, and quality problems led to rework
- New determinants for productivity and delays
 - Stability of markets and organization
 - Process consistency with product characteristics
 - Support for collaboration of organization
- Tooling and team sizing seemed unproblematic

Summary and Conclusions

- Problem:
Method misalignment leading to productivity problem and delays
- Approach:
Inductive, bottom-up research approach
- Results:
Challenges with their causes and impacts in a large-scale agile development
- Contribution:
Problem description for process improvement
- Next steps:
Theory-building and solutions