Agile Research into Agile Development

James Noble
Victoria University of Wellington
New Zealand

And a cast of thousands

Agile Research into Agile Development

- Agile Development
- Grounded Theory
- Research Results
- Programming as Theory Building

Extreme Programming - 1999

Kent Beck

Ward and Kent – 1980-1990s

Smalltalk
Agile Manifesto - 2001

Agility

“the ability of an organization to react to change in its environment faster than the rate of these changes”
Steve Adolph
AGILE 2006

XP Practices

Agile Beliefs

- “Scrum works in any environment and can scale to programming in the large”
  Jeff Sutherland, Retrospective on SCRUM and its Implementation in Five Companies, Project Keeper, Inc., 2001.

- “Agile Architecture IS Possible — You First Have to Believe!”
  Mark Isham, AGILE 2008

a developer rebellion against unwanted tasks and schedules ... an opportunity to sell ... certification and training;
voice inc, The Agile Dilemma, quoted by Dave Ramel, Application Development Trends (adtmag) 2012

Grounded Theory

The Discovery of Grounded Theory

- A research method to study human aspects and social interactions in situ
- Captures the main concern of the participants
- not the researcher’s main interest
- Complete research method — not just analysis
A Grounded Theory Project

- Qualitative research
- 28 participants, 23 organizations, 3 years
- New Zealand and India
- Started with area of interest:
  Agile Project Management
- Semi-structured Interviews and Observations
- Data analysis
- Research Focus: Self-Organizing Agile Teams

Open Coding

“Big, big issues are getting enough collaboration time with the [customers]… there’s no way less than a full time person would be able to keep up with getting all the requirements.” P14, NZ Coach

Open Coding

“Big, big issues are getting enough collaboration time with the [customers]… there’s no way less than a full time person would be able to keep up with getting all the requirements.” P14, NZ Coach

Open Coding

“Big, big issues are getting enough collaboration time with the [customers]… there’s no way less than a full time person would be able to keep up with getting all the requirements.” P14, NZ Coach

Open Coding

“Big, big issues are getting enough collaboration time with the [customers]… there’s no way less than a full time person would be able to keep up with getting all the requirements.” P14, NZ Coach

- Lack of customer collaboration time (P14)
Codes to Concepts

The
- Lack of customer collaboration time (P14)
- Insufficient time allocated to customer representatives (P11)
- Playing Agile customer alongside operational job (P5)

Time Commitment

Growing the Research

- Theoretical Sampling:
  - Guides choice of subjects & Content of interviews
- Memoing:
  - Write about codes, categories, and relationships
- Sorting:
  - Organise Memos and Ideas

Concepts to Categories

Skepticism and Hype
Distance Factor
Time Commitment
Large Customers
Negotiating Contracts
Ineffective Customer Rep.

Core Category

One of the categories emerges as core
- "Self Organising Agile Teams"
- Selective Coding
- Theoretical Saturation
- Theoretical Coding Families

Theory

Process
- Research Area
- Minor Literature Review
- Data Collection
- Open Coding
- Constant Comparison
- Memoring
- Code / Concept / Category
- Core Category
- Selective Coding
- Sorting
- Theoretical Saturation
- Theoretical Coding
- Write-up

Research Results
- XP Customers
  - Angela Martin


XP Practices
- Morning Standup
- Pair Programming
- Code Reviews
- Architecture / Stakeholders
- BDD / Specification
- Defects
- Testing
- Refactoring
- Continuous Integration
- Formal Reviews
- User Stories
- Writing
- Practice

What is it like to be an XP Customer?
"I think it’s worked very well, but ... I don’t know how long I can keep this pace up"
— Product Owner, BigLotCorp

"Overall — I love this approach ... and I’d certainly like to use it again ... We probably needed about three of me ... It’s been my life ... Look at these grey hairs"
— Product Owner, ABCCorp

"I’ve always worked at least 70 or 80 [hrs a week] I don’t even mind, it’s like what I do"
— Product Owner, RandomCorp

The Theory of an XP Customer (Team)
- 16 Roles
- 8 Practices

Roles
- Developer
- Tester
- Customer
- Team
- Manager
- Architect
- Business Analyst
- Scrum Master

Role: Job Title
A role can have just one person doing it or many people doing it. A person can do just one role or may be doing multiple roles.
**Technical Liaison**

"...different teams... always comprise a project, you’ve got the UI, you’ve got deployment, there’s getting the applications out, initiatives etc., you’ve got security, you’ve got infrastructure...

...the interaction between different systems, so you’ve got all these different groups that you bring together. Customer, SkillCorp.

Product Owners who attempt to deal with all of the technical liaison quickly become overloaded. This task deserves a person to be responsible for it.

---

**Political Adviser**

“Well we knew that if we actually got people’s formal sign-off, business people sign-off for everything, we’d never actually get anything done. So we ran a bit of a risk. We assumed, based on [senior executive]’s work with people and our knowledge that we could proceed without formal sign-off.”

Product Owners are often project focused and so cannot be involved in organization politics. Product Owners need help to identify players and rules:

- Who needs to say “yes”. Who needs to say “no”?
- Which rules to follow, which roles to break.

---

**Practices**

- Real Customer Involvement
- Whole Team
- Energized Work
- Customer Appreciation
- New User Feedback

**Programmer On-Site**

Programmers need to understand and respect their users:

Very old advice – get the programmers into the field:

- “Being off site is like a programme session on an early flight.”
- “I worked with a social worker; doing a death review. That’s what she does every day.”

Not about making decisions:

- Understanding the end user and context of use
- Making your suggestions actually helpful

---

**Programmer Holiday**

Agile projects can be very intense:

Product Owners sometimes need some time to get ahead of programmers:

- The day ahead dynamic is REALLY important

Product Owners sometimes need to send the programmers on holiday:

- For example, an iteration focused on technical refactoring and for technical debt

---

**Research Results**

- Self-Organising Teams
- Outsourced
- Rashina Hoda


XP Practices

Self-Organizational Roles

Mentor
Terminator
Promoter
Co-ordinator
Translator
Champion

Mentor

"With the Mentor, I would often teach what's new in Agile work and shape our mindset and make sure everyone is involved in work under the Agile umbrella."

Developer, NZ

"You need to show them, a client presentation [that] worked, and after a few such validation cycles, they start to get confident."

Agile Coach, NZ

Champion

"The organization was getting the most benefit from Scrum. We organized a few senior management meetings to get them on board."

Scrum Team, India

"The [Champion] was pretty much championing the whole Agile idea. "Senior Management" were thinking of using the Champion to expand Agile throughout the organization."

Business Analyst, NZ

Terminator

"An [team member] took the whole product and gave a presentation to the team. They showed us a few hundred percent complete elements."

Business Analyst, NZ

"Some stakeholders have 100 percent complete elements. We examine as a Coach. The worst areas receive sometimes hard reality. People get frustrated. [Agile] identifies the good things [Agile] Jensen even the bad things."

Agile Coach, India

Summary

- [Green checkmark]
- [Green checkmark]
- [Green checkmark]
- [Green checkmark]
- [Green checkmark]
- [Green checkmark]
- [Green checkmark]
Research Results

- Globally Distributed Teams
- Siva Dorairaj

Siva Dorairaj, James Noble, Petra Melik. Understanding Team Dynamics in Distributed Agile Software Development. XP 2012.

XP Practices

Share the Pain

“We alternate sprint planning meetings between mornings and evenings for both sites. It’s fair in the sense that it is convenient to one site but inconvenient to another, and we rotate it for every sprint. We share the pain, not just the glory.” —P46, Agile Coach.

Theory of “One Team”

Planned Unconference

“Before retrospectives [meetings], we make sure that there’s always time allocated for casual conversation. Somebody will say, “My son got an award”, or “I went on a vacation and I did hang-gliding” —P45, Agile Coach.

Know the Team

“There’s a really simple way to know the faces of our team in Beijing, Bangalore, or Melbourne. In all those offices, we have nice, big photographs of each of the team members up on the wall so that we can see the whole team.”
—P22, Scrum Master.

“If I didn’t have the photographs, I’d draw little characters for them…” —P2, Agile Coach.
Research Results

- Agile Architecture


XP Practices

[Image of XP Practices diagram]


Agile Architecture

- Unstable requirements
- Early value
- Team experience
- Non-agile environment

Agile architecture design

- Full up-front architecture
- Address the technical challenges
- Plan for options
- No up-front architecture

Strategies

Vendor Frameworks

“...You choose the proper plug-ins and then you get the functionality that you are looking for.”
(P16, CEO)

“Those architectural decisions can be very emergent nowadays; I don’t think they’re nearly as intractable”
(P29, development manager).

Or not...

“There were a number of architectural things that were developed in-house. [...] We had a bit of a go with what Microsoft had off the shelf previously, found it painful and limiting, and felt that it confirmed our decision to go our own way with data binding.”
(P13, architect)

Size vs Complexity

“If we have size that just extends the time, it’s of little concern to us. It’s just a slightly larger backlog management overhead.”
(P32, software development director)

“the complexity of an organisation’s systems ... has a greater influence on the amount of forethought required than the budget or size”
(P10, agile coach)
How can you trust a GT study?

- Subjectivity vs Generalisation
  - Should describe how the study was carried out
  - Should "accurately describe the context studied"
  - Should not generalise outside those contexts

- Confidentiality vs Openness vs Replication
  - Give access to raw data!
  - Cannot replicate by "re-analyzing" data
  - Can replicate by repeating a similar study
  - Test generalisation by repeating a different study
  - Theory may generate testable propositions

Challenges & Limitations

- Effort
  - Full GT study: 40-60 participants
  - Multiple projects. Multiple visits/observations.

- Grounded Theory Community
  - Glaserites vs Straussians

- Acceptable to SE Programme Committees

- You can do bad research with any method

Grounded Theory is Agile!

- Start with a general Research Area
- Data collection begins early
- Data analysis begins early
- Publication begins early
- Research Proposal includes preliminary results
- Research Questions emerge along with results
  - (but they do in most research projects)
- Data drives Theory drives Data drives Theory...

Novelty & Surprise

"We all know that already..."

- in a user group (or conference...)
- from a single project
- of a few projects
- over several projects
- over many projects
- over many projects
- from lab/classroom/simulation

Agile is Grounded Theory?

- Peter Naur, Programming As Theory Building
  - Reprinted as Appendix B of Alistair Cockburn's Agile Software Development.

"The theory built by the programmers has primacy over such other products as program texts, user documentation, and additional documentation such as specifications."
Development as Theory Building

- Analysis / Development techniques (not just Agile) are similar to GT techniques
  - Interviews — Conversations with the Customer
  - Open Coding — Story Writing & Textual Analysis
  - Theoretical Sampling — Story Planning
  - Constant Comparison — Refactoring
  - Iteration
  - Traceability
  - Feedback from real "customers"

Agile Research into Agile Development

- Individuals and interactions over processes and tools
  - "all is data" — Individuals > Interactions > Processes > Tools
  - Working software over comprehensive documentation
  - Working theory over comprehensive descriptions
- Customer collaboration over contract negotiation
  - Participants main concerns over research questions
- Responding to change over following a plan
  - Theoretical sampling over preselected participants

Agile Research into Agile Development

- Agile Development
  - Agile "works"
    - But how? why? Always?
- Grounded Theories of Agile Development
  - Rich, qualitative descriptions
  - Practically useful contributions
- Agile research methodology

Any more questions?

Further Reading

- Confessions from a Grounded Theory PhD: Experiences and Lessons Learned. Dominic Furniss, Ann Brandford, Paul Curzon. CHI 2011