



Agile project management: A magic bullet?

Prof. Darren Dalcher

d.dalcher@mdx.ac.uk

Conferencia Iberoamericana de Calidad del Software



Outline

- I. What is agility?
 - *The agile manifesto*
- II. Agile methods sampler
- III. The claims: Does it work? Do we know?
 - *Findings*
 - *Discussion & implications*
- IV. Agile project management
 - *CSF*
 - *Problems and limitations*
- V. Open questions



Part I: What is Agility?

- According to dictionary:

Agile:

nimble,
quick moving,
active,
mentally quick



Agility is every where

- Agile web/software/product development
- Agile project management
- Agile estimating and planning
- Agile modelling
- Agile documentation
- The agile investor
- Agile competitors
- Agile systems/organisations/virtual enterprises
- Agile energy systems
- Creating the agile library
- The agile gene



New terminology

- Sprint
- Velocity
- Scrum
- Extreme programming
- User stories
- Epics
- Sagas
- Pair programming
- User retrospectives
- Huddles
- Daily stand-up meetings



Agile methods

- No agreement of what the concept of 'agile' actually means.
- One can refer to the 'agile movement' or the 'agile school'.
- Common thread is belief in principles.
- Belief that requirements are difficult for users.
- Users cannot articulate or define and may not even know what they want.
- Partial delivery can stimulate more and better requirements (in evolutionary fashion).



Reflection: the problem

- How do you speed up delivery on a project?



How do you speed up development?

- Put more people on the team
- Cut corners
 - *Less documentation*
 - *Less testing*
 - *Less quality*
- Pressurise the team
 - *Work longer*
 - *Work weekends*
- Buy tools and methods to help speed things up
- Deliver less



The answer

Change the way we do things:

- Involve the users
- Set up small teams
- Establish partnerships
- Communicate
- Deliver basic functionality quickly



The agile manifesto (values)

Individuals and interactions

over processes and tools

Working software

over comprehensive documentation

Customer collaboration

over contract negotiation

Responding to change

over following a plan



Agile principles I

- Highest priority - satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development
- Deliver working software frequently
- Business people and developers must work together daily



Agile principles II

- Build projects around motivated individuals and give them support
- Face-to-face communication is the best method for conveying information
- Working software is the primary measure of progress
- Promote sustainable development



Agile principles III

- Continuous attention to technical excellence and good design
- Simplicity is essential
- The best work emerges from self-organising teams
- The team regularly reflects on how to become more effective, then adjusts its behaviour accordingly



Part II. Agile methods sampler



Many flavours of agile

- XP: Extreme Programming
- SCRUM
- DSDM
- ASD: Adaptive Software Development
- FDD: Feature-Driven Development
- Crystal
- Agile Modelling
- Agile Unified Process
- Software Development Rhythms
- Agile Data Method



XP: Extreme Programming

- Widely used and well publicised
- Team of 5-10 programmers in one location
- Development in builds delivered incrementally
- Requirements specified as user stories
- Programmers work in pairs
- Do their own testing
- Requirements, architecture and design emerge over course of development
- Customer integral to team or accessible
- Innovative & controversial
- All about programming

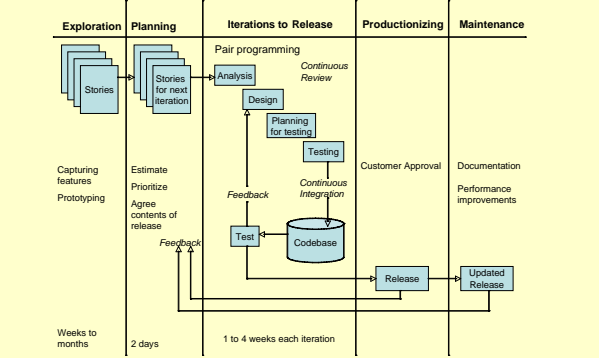


Extreme Programming innovative practices

- Small releases (two weeks)
- Create test before coding
- Refactoring
- Collective ownership
- Continuous integration
- Sustainable pace
- On-site customer (full time)
- Pair programming



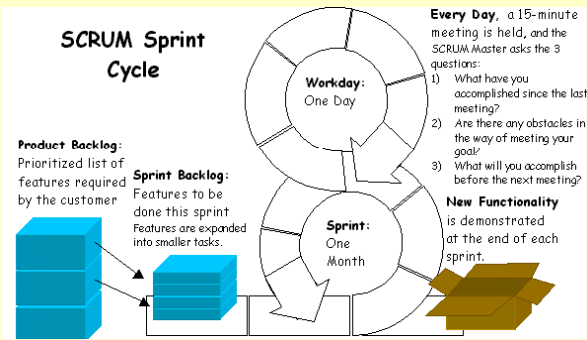
The process for Extreme Programming



SCRUM:

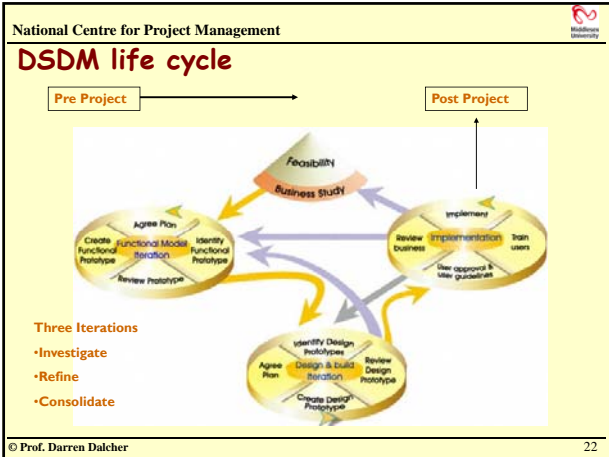
- Sprints: iteration of 30 day duration
- Work within each sprint is fixed
- Then, only the team can add new functionality
- Work to be done characterised as product backlog
- Mentored and managed by a SCRUM Master
- Self-organised, self-accountable teams
- **Daily stand-up meeting**
- **Time boxing** is key emphasis
- Guidance calls for three sprints per release
- So, market release on a 90 day timeframe


The SCRUM life cycle




DSDM: Dynamic Systems Development Method

- Requirements cannot be (truly) fixed
- Small proportion of requirements, will deliver the bulk of value to user
- Nothing is built perfectly the first time
- Date and resources are fixed so must change requirements
- Series of **time-boxes**
- MoSCoW prioritisation
- Originally a RAD so supports **prototyping**
- Developed by consortium
- Covers development and project management



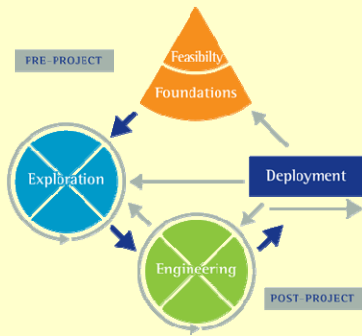
- National Centre for Project Management 
- ## DSDM overview
- ❖ Based on an 'Agile' approach
 - ❖ *Not about taking short cuts*
 - ❖ Works on the principle that nothing will be built perfectly first time, and plans for this
 - ❖ Uses the Pareto Principle (80:20)
 - ❖ *Can produce a usable and useful 80% of the system in 20% of the time*
 - ❖ Independent of any particular tools and techniques
 - ❖ Facilitated workshops are a key activity to allow communication
- © Prof. Darren Dalcher 23

- National Centre for Project Management 
- ## MoSCoW Prioritisation
- ❖ **Must have** minimum useable subset
 - Essential - major business benefits
 - ❖ **Should have** very important, but not Musts
 - Ought to have - good benefits
 - ❖ **Could have** less important than Should
 - Nice to have - some benefits
 - ❖ **Wont have** this time round, but may have later
 - Maybe next time ...
- © Prof. Darren Dalcher 24

DSDM particularly suited to projects with:

- Empowered and skilled teams
- Committed and involved end users
- Accommodation of frequent delivery
- Development teams of 6 people or fewer
- The ability to create prototypes
- Highly demonstrable user interfaces
- Fixed timescales, flexible requirements
- Computationally non-complex requirements

DSDM Atern life cycle



Part III: What do agile methods achieve?



Agile survey

2006, Scott Ambler, 4232 IT professionals:

- 65% of organisations have adopted one or more agile techniques
- 41% of respondents work in organisations who use an agile method
- 60% show increased productivity
- 66% show increased quality
- 58% show increased stakeholder satisfaction

Concerns: self selecting group?



Experiment: The Rationale

Should we use eXtreme programming instead of the Waterfall?

Issue: Various life cycles on offer

Required: Informed decisions about which one to adopt

Problem: No access to real comparative data

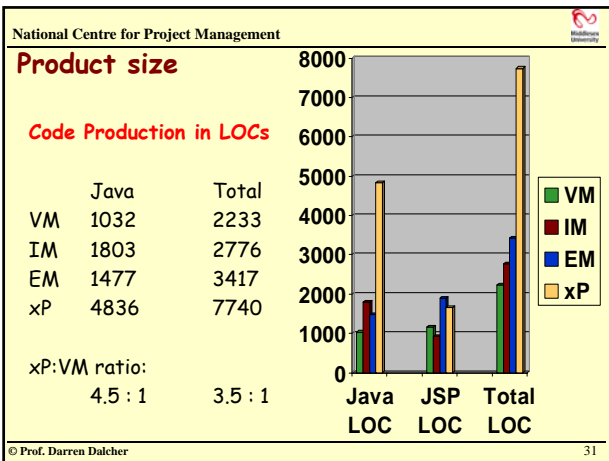


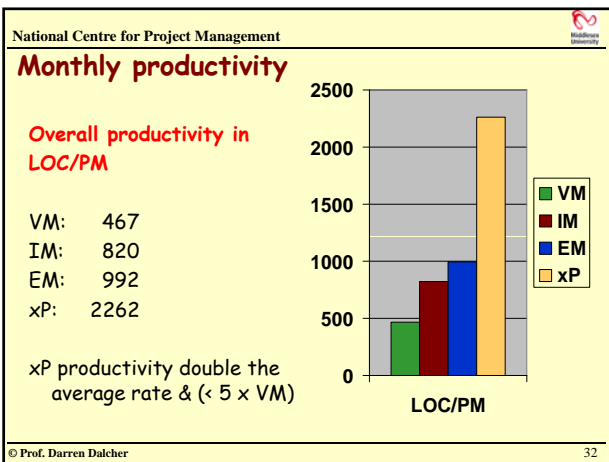
The Experiment

Comparison of four development approaches

- The V-model
- Incremental development
- Evolutionary development
- eXtreme Programming

in terms of the product, attributes & project





National Centre for Project Management

Assessment

Most comprehensive solutions,
resulting in highest level of satisfaction from
the users:

xP

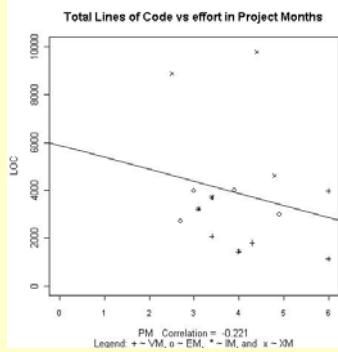
© Prof. Darren Dalcher 33

Size as a function of effort

for same effort
 xP greater
 VM worst

Boehm et al.:
 Development effort is generally proportional to size of developed Product

But xP in different league



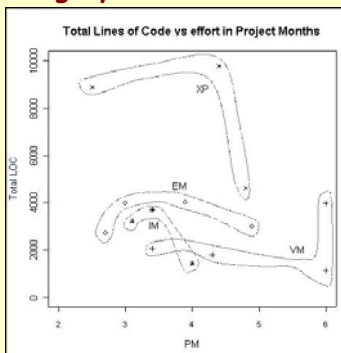
Productivity clustering by method

xP more productive

Medium size:
 Boehm: 8-32K
 Fairley: 5-50K

xP teams developed medium products in same time

Other teams produced small products



Productivity rates

Industrial productivity:
 200-500 LOC/PM

Kan reports:
 2.8-6 classes/PM

VM: 467

VM: 1.7

IM: 820

IM: 5.9

EM: 992

EM: 7.4

xP: 2262

xP: 7.1

All but VM exceed upper range

All but VM come close or outperform



Change of paradigm?

- Programmers spend small proportion of their time programming (Fairley 1985) 13-15%
- xP tries to maximise time available for programming (e.g. minimal documentation)
- Result: enhanced output (more code, more screens) delivered more rapidly



Open questions

- Is XP a life cycle?
- Does it represent a new paradigm?
- We were hoping to focus on quality using ISO9126. How do we measure quality?



Reflections: Where does XP fit in?

- XP, the teams meant to be 'working in the small' delivered the biggest and most comprehensive products
- IM and EM were offered as alternatives to VM-type approaches
- XP seems to be radically different to other solutions that came before it
- (Essence rather than accident?)



Part IV: Managing projects the agile way

It's project management but not as we know it!




Agile management

- Agile methods designed for things we cannot control
- Agility: the ability to thrive and prosper in an environment of constant and unpredictable change
- Not to accommodate change; but to relish it!
- Opportunities can thus open up in a turbulent environment
- (You wouldn't want to freeze your concept in a changing market).

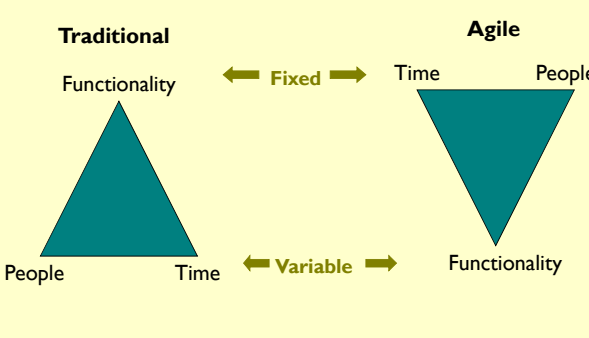


CSFs for implementing agile practice


- Open organisational culture
- Readiness for mind shift
- No fixation on completing requirements
- Team size smaller than twenty
- Correct environment to support agility principles
- Adoption of all principles in one go
- Availability of an experienced coach
- Opportunity to communicate

National Centre for Project Management 

Traditional vs. agile Projects




© Prof. Darren Dalcher 43

National Centre for Project Management 

How is agile PM different?

- Timeboxing (in days or weeks)
- Fixed resource envelope
- Delivering value vs. delivering projects
- Short-term focus
- Optimise locally at team (microproject) level

© Prof. Darren Dalcher 44

National Centre for Project Management 

Implications

- Becomes routine delivery (→ hence not project work???)
(→ work as normal or task management???)
- Serious contractual implications
- → greater obsession with process
- Extreme can take the balance too far

© Prof. Darren Dalcher 45



Problems with Agile methods

- License to hack?
- What happens in outsourcing and off shoring?
- Safety-critical environments?
- Will the user be willing and able to spend time with the team?
- Right personalities within team?



Where agile will not work (yet)

- Larger products and teams
- Distributed environments
- Safety-critical systems
- Stable environments (rework could be too expensive)
- Experts and key personnel not available
- Users are not available
- Highly ordered environments
- Cultures not supportive of freedom, trust, openness, responsibility, empowered decision making and taking charge of the future



Part V: Some concerns and open questions



My big concerns regarding agility...

- Many unsupported or unproven claims
- Encourages people to ditch some of the ideas on the right (e.g. processes, documentation)
 - always a question of balance.
- Portfolio of projects built bottom-up
- Prioritising changes of multiple stakeholders is difficult (whilst losing overall picture)
- Supposed to be good at handling risks, but little knowledge or direction



My big concerns regarding agility...

- Requirements document basis for traditional contract. New form of contract needed that pays for time rather than outputs...
- Greater local autonomy (to optimise micro-project) may impact global efforts
- Maintaining simplicity will require extra infra structure work
- Managing task means shifting priority and focus of PM
- Who sees overall picture?



Open questions

- Antidote to bureaucracy or license to hack?
- Focused microprojects by SWAT teams?
- Is this what a truly focused project should look like?
- Agile was developed to respond to change and emphasise the role (and diversity) of people, but does it only deal with change-in-the small?
- Can teams prioritise (commit to and select) work without having a higher authority?
