MSc Information Systems

Agile Development Methodologies

Module SITS code: COIY059H7
Dr Brian Gannon

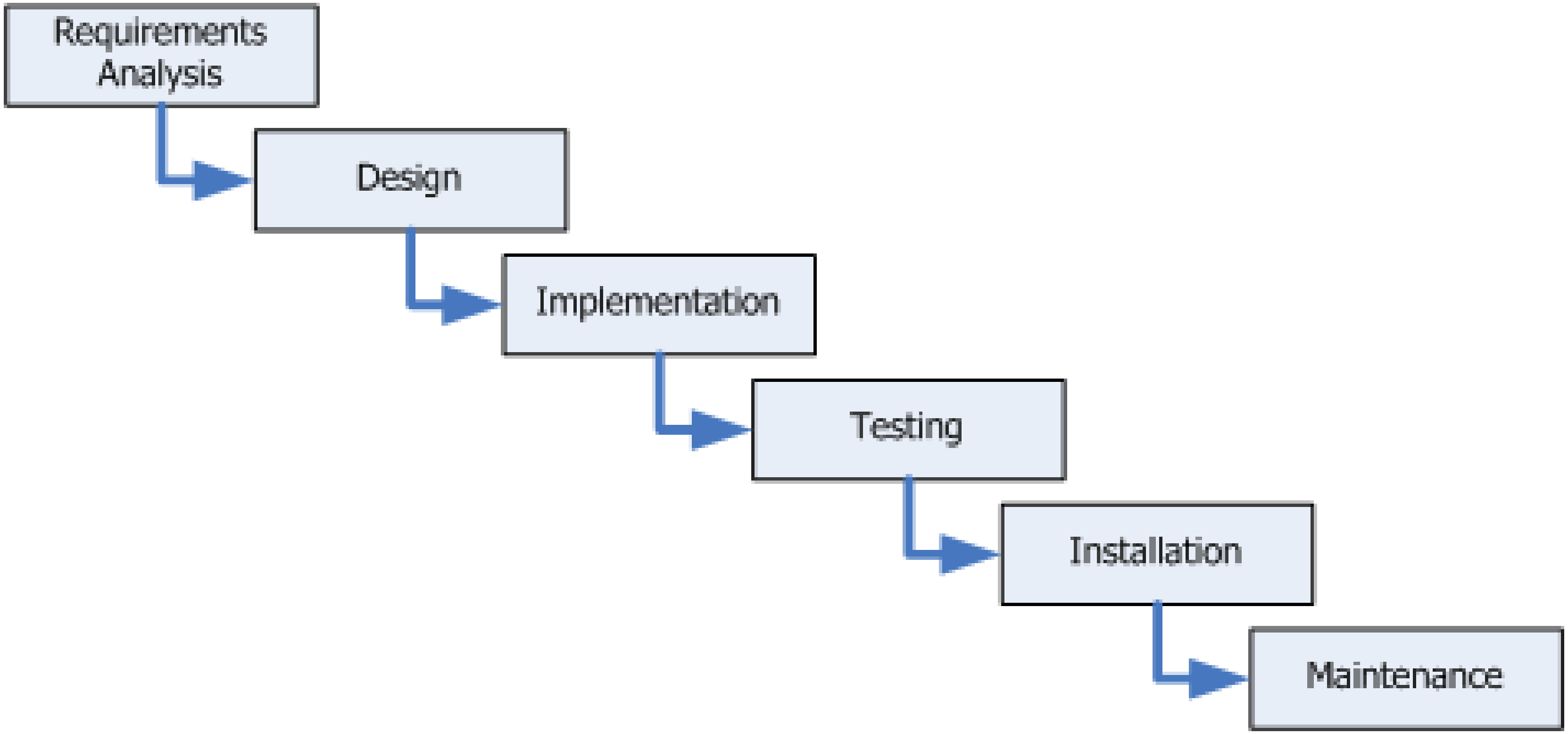
Systems Development Methodology

- A methodology is a formalised approach to implementing the SDLC

- A methodology can be:

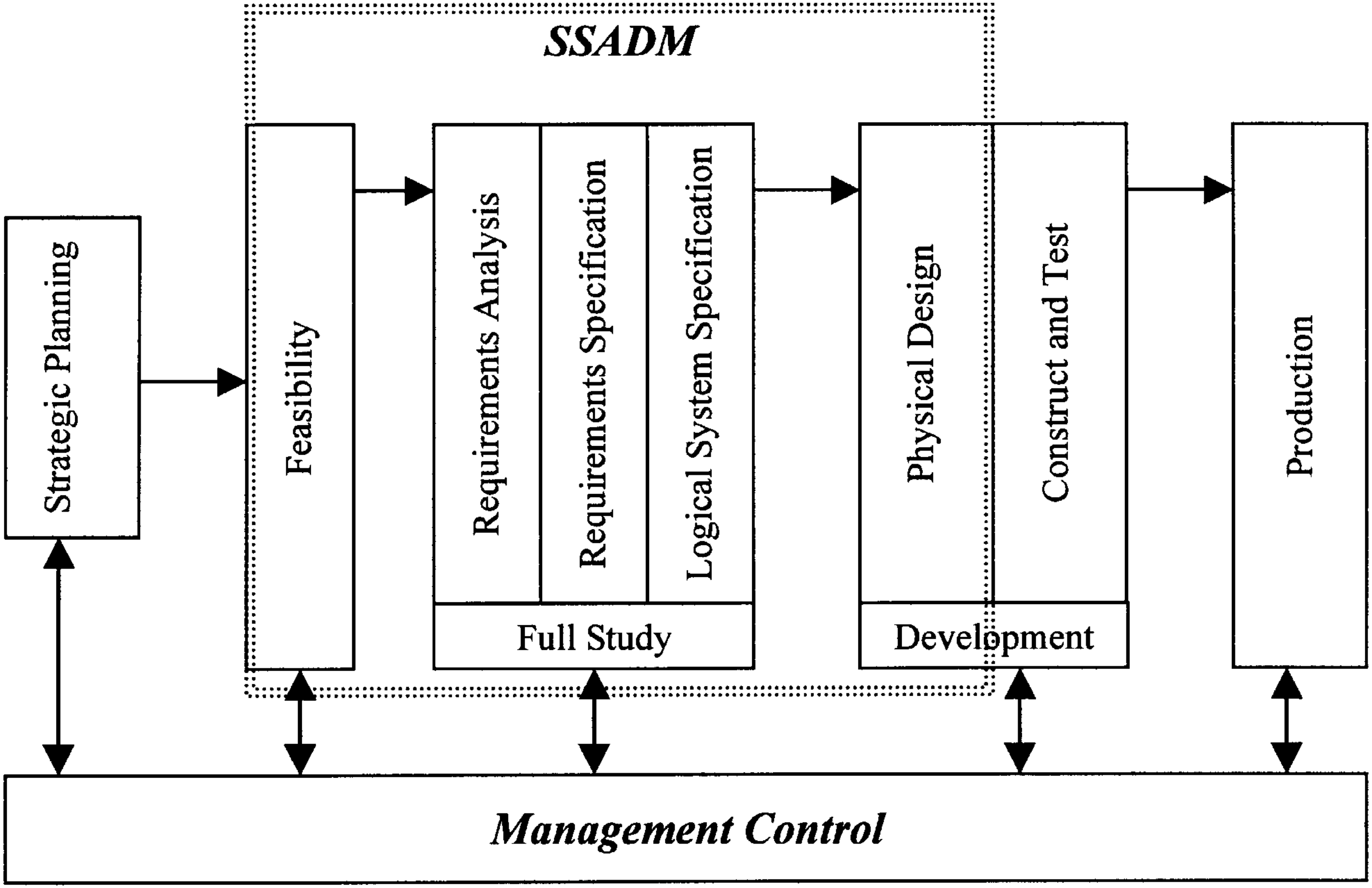
  - Process-oriented
  - Data centered
  - Object-oriented
  - Structured
  - Agile
  - Other….

Waterfall Methodology

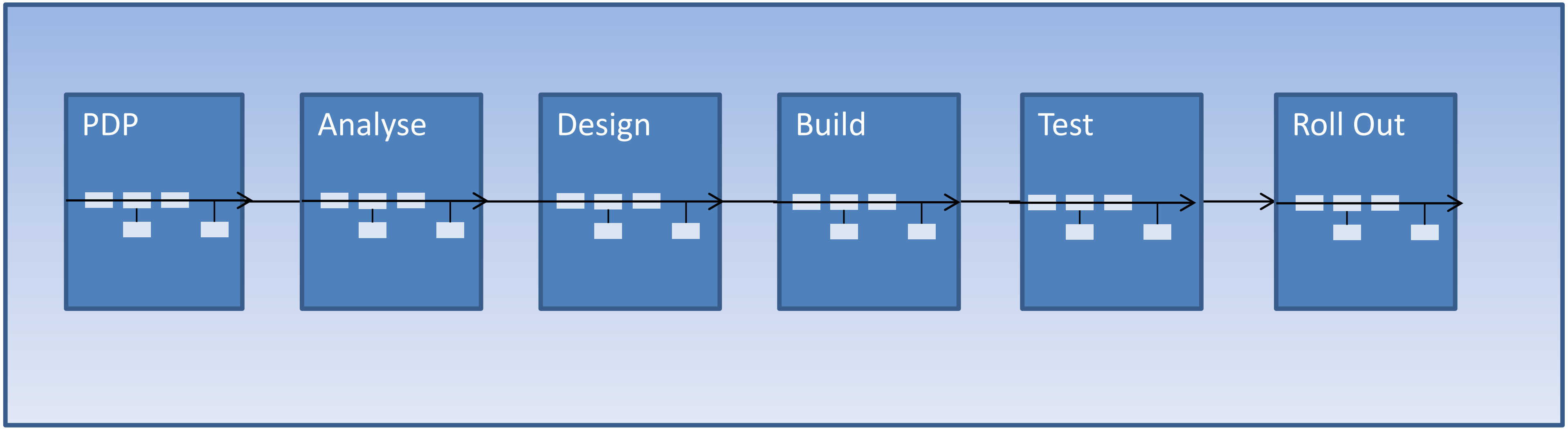


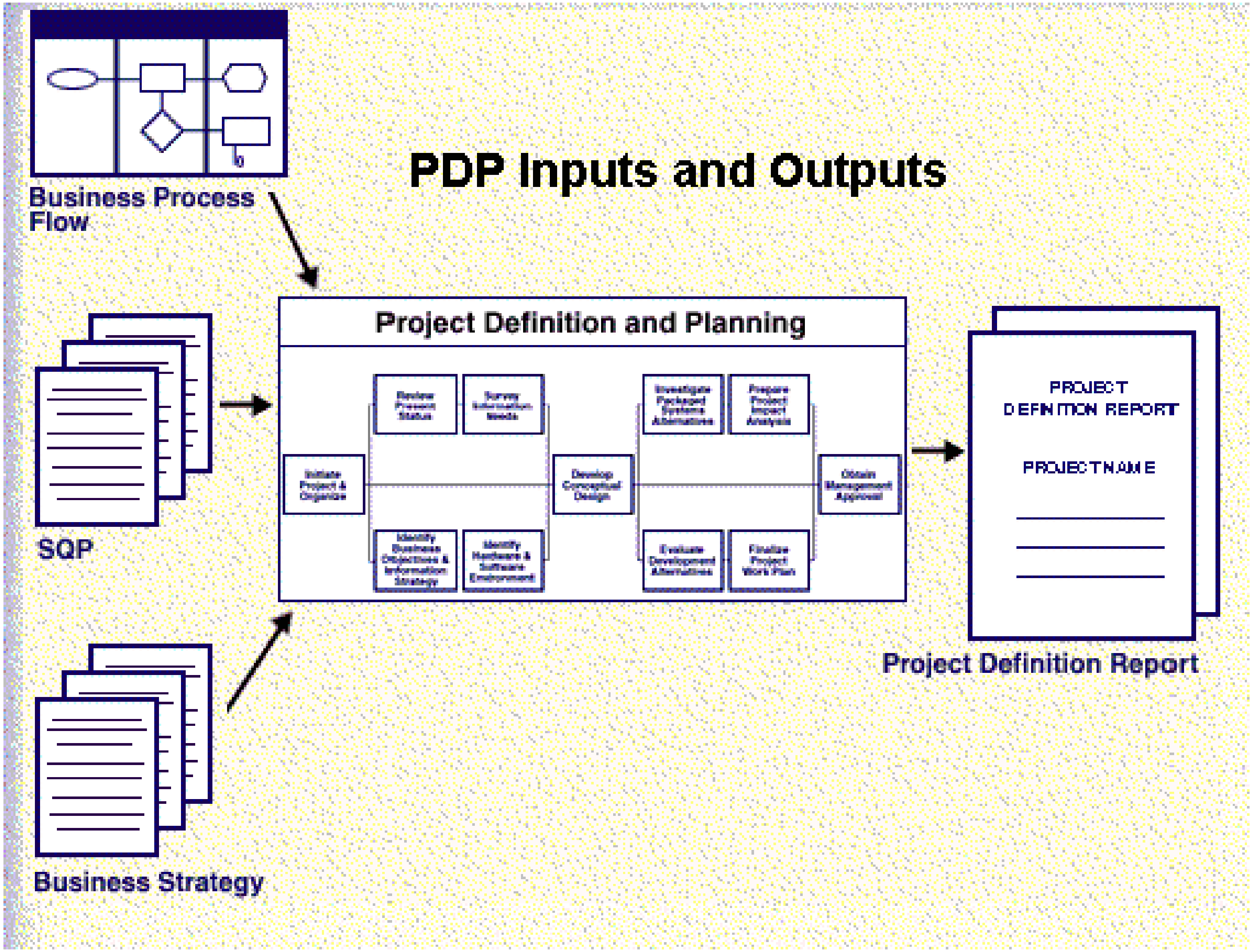
http://www.projectsmart.co.uk/waterfall-v-agile-how-should-i-approach-my-software-development-project.html

SSADM


SSADM
Strategic Planning
Feasibility
Requirements Analysis
Requirements Specification
Logical System Specification
Full Study
Physical Design
Construct and Test
Development
Production
Management Control

METHOD/1

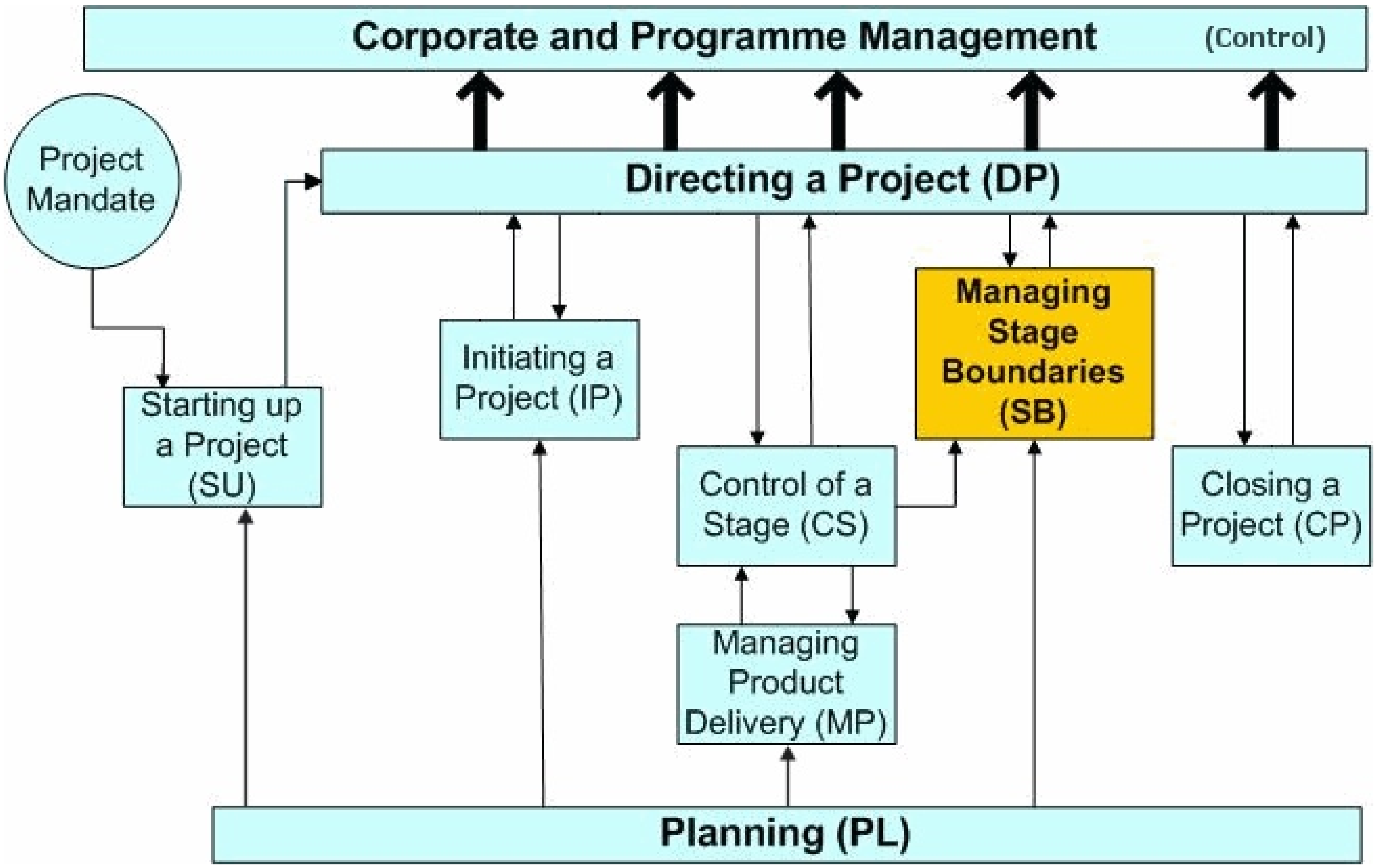




PDP Inputs and Outputs

Business Process Flow
SQP
Business Strategy
Project Definition and Planning
Initiate Project & Organize
Review Present Status
Survey Information Needs
Identify Business Objectives & Information Strategy
Identify Hardware & Software Environment
Develop Conceptual Design
Investigate Packaged Systems Alternatives
Prepare Project Impact Analysis
Evaluate Development Alternatives
Finalize Project Work Plan
Obtain Management Approval
PROJECT DEFINITION REPORT
PROJECT NAME
Project Definition Report

PRINCE/2


Corporate and Programme Management
(Control)
Project Mandate
Directing a Project (DP)
Starting up a Project (SU)
Initiating a Project (IP)
Managing Stage Boundaries (SB)
Control of a Stage (CS)
Closing a Project (CP)
Managing Product Delivery (MP)
Planning (PL)

http://upload.wikimedia.org/wikipedia/en/b/bc/Prince2_procces_model_.jpg

Waterfall Criticisms

Pros	Cons
Detailed specification of requirements	Users rarely know all requirements up front
Precise plan and road-map	Limited visibility of product until late stage
Controlled and prescriptive - suitable for less experienced developers	Inflexible – difficult to change direction or requirements
Defined start and end points, so easier to measure progress	Much effort and planning needed before starting
Clear demarcation at each phase, and between development and operation	Creates distance between developer and user

Agile Manifesto

Agile Manifesto

- **Individuals and Interactions** over *processes and tools*
- **Working software** over *comprehensive documentation*
- **Customer collaboration** over *contract negotiation*
- **Responding to change** over *following a plan*

agilemanifesto.org

Agile Principles

“The Agile movement is not anti-methodology, in fact many of us want to restore credibility to the word methodology. We want to restore a balance. We embrace modeling, but not in order to file some diagram in a dusty corporate repository. We embrace documentation, but not hundreds of pages of never-maintained and rarely-used tomes. We plan, but recognize the limits of planning in a turbulent environment.”

— Jim Highsmith, History: The Agile Manifesto

Agile Principles

1. Customer satisfaction by rapid delivery of useful software
2. Welcome changing requirements, even late in development
3. Working software is delivered frequently (weeks rather than months)
4. Working software is the principal measure of progress
5. Sustainable development, able to maintain a constant pace
6. Close, daily cooperation between business people and developers

Agile Principles

7. Face-to-face conversation is the best form of communication (co-location)
8. Projects are built around motivated individuals, who should be trusted
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximising the amount of work not done—is essential
11. Self-organising teams
12. Regular adaptation to changing circumstances

Agile Objectives

1. Continuous Innovation

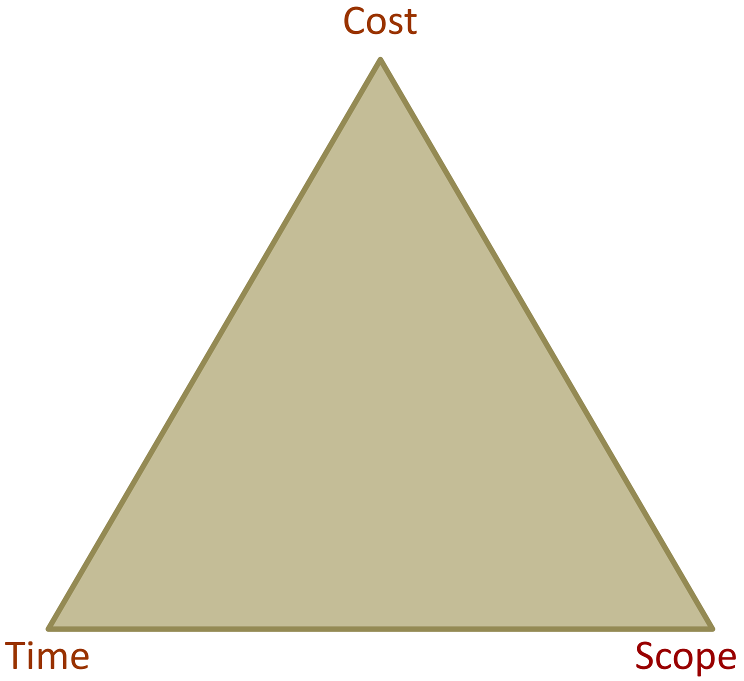
2. Improved time-to-market

3. People and process adaptability

4. Product adaptability

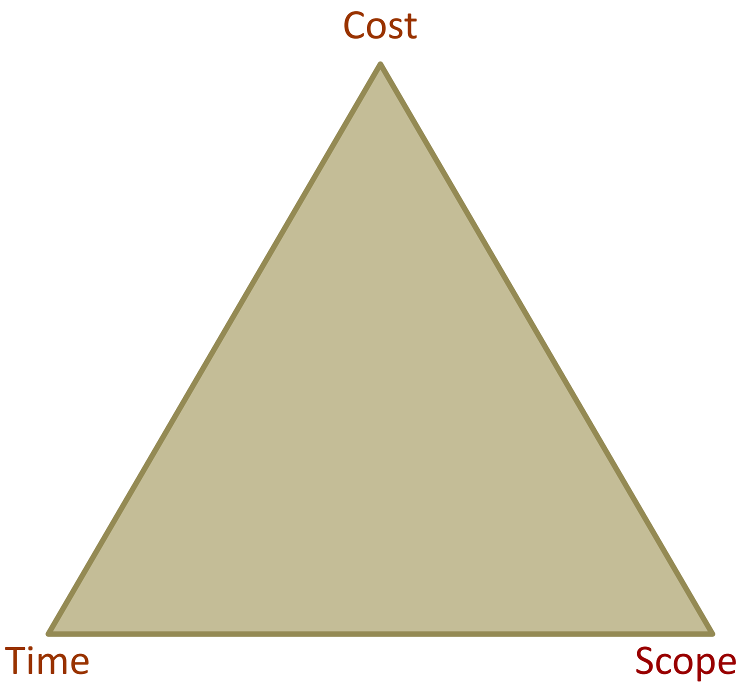
5. Reliable results

Traditional Project Trade Off

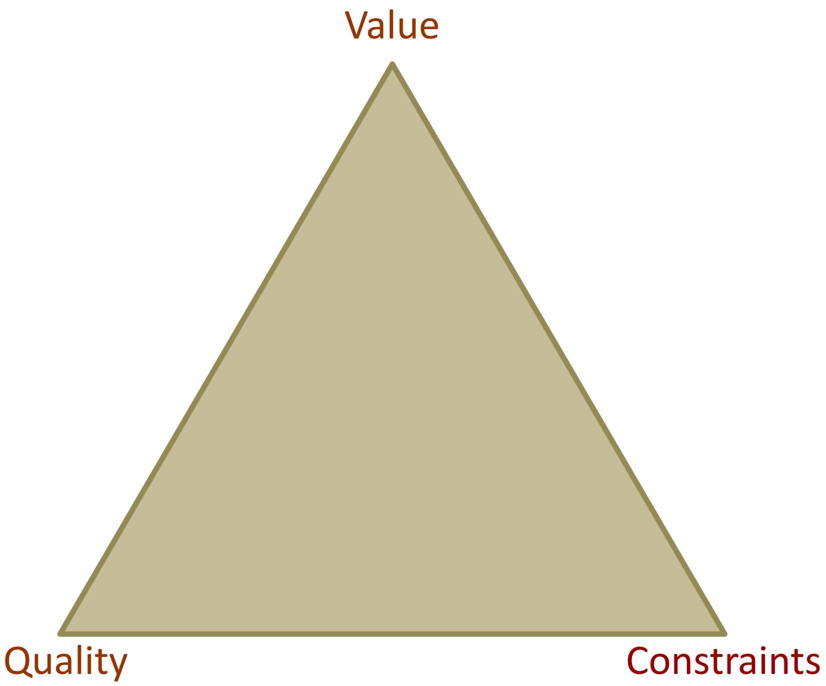


traditional trade off

Traditional Project Trade Off

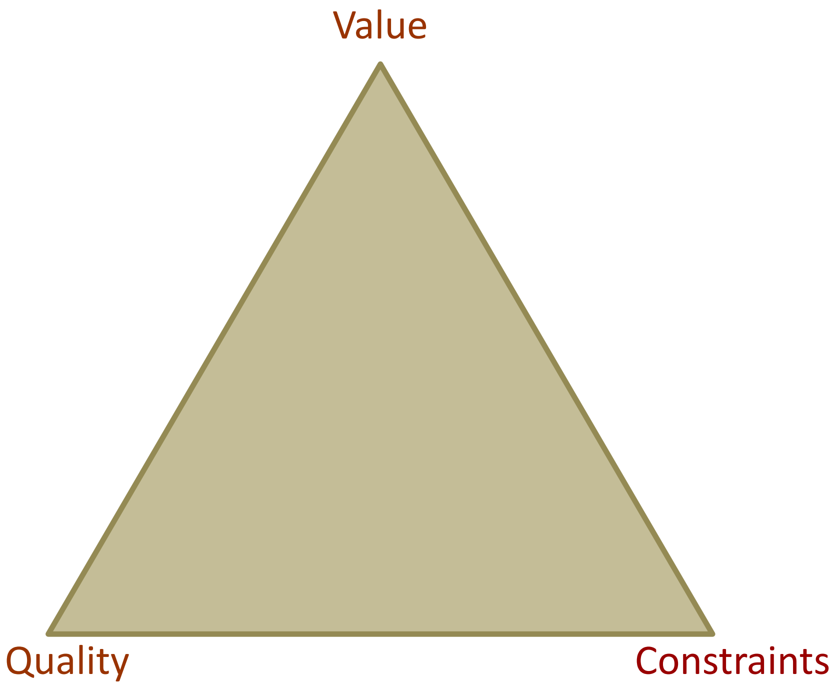


traditional trade off



agile trade off

Agile Goals



agile trade off

Value Goal: Build a releasable product

Quality Goal: Build a reliable adaptable product

Constraint Goal: Achieve Value and Quality goals within acceptable constraints

“Agile Project Management”, Highsmith, 2010, p.21

Agile Project Management

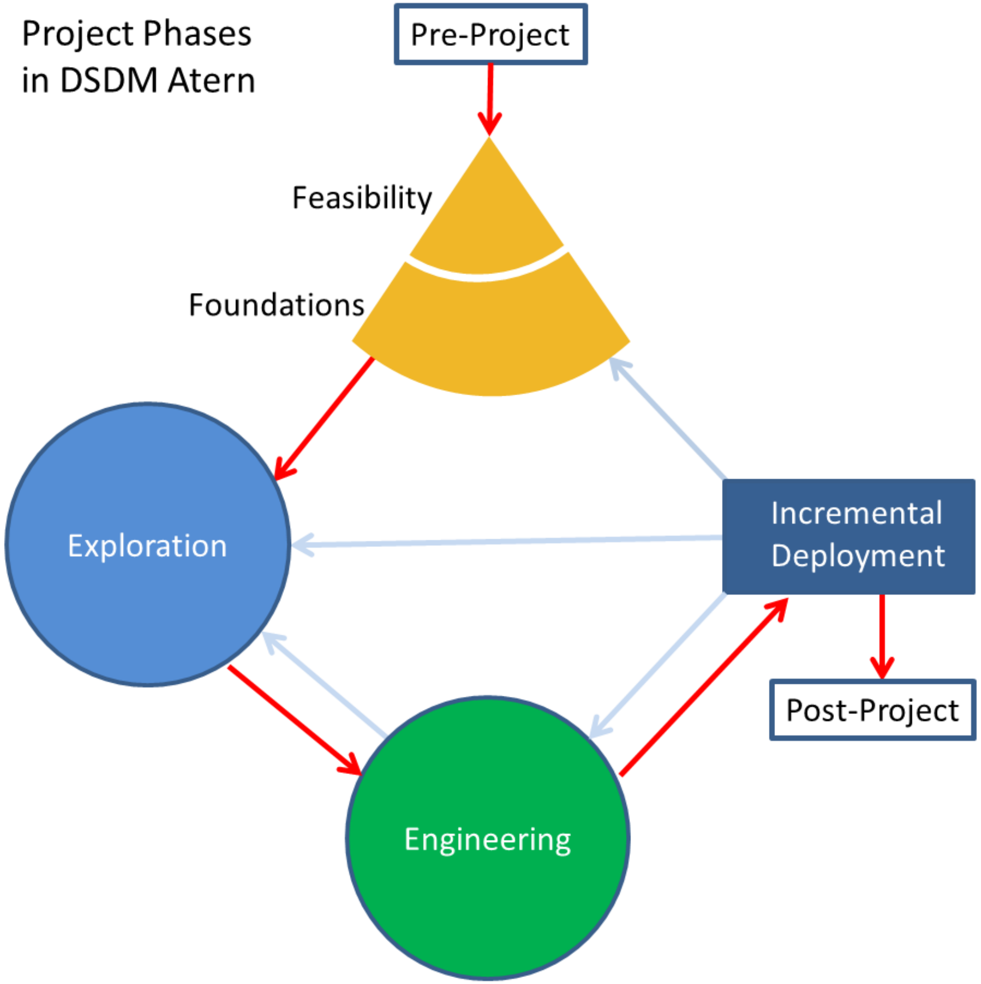
- Agile assumes development teams consist of knowledge workers
- Agile provides six practices for managing projects
  - Guiding vision
  - Teamwork and Collaboration
  - Simple Rules
  - Open information
  - Light touch
  - Agile Vigilance

- Agile requires clarity around:
  - Business Needs
  - Business Agreement
  - Project Sponsor
  - Business Team Members
  - Team Size
  - Tools & Techniques
  - Prioritisation

Dynamic Systems Development Method (DSDM) Approach

- www.dsdm.org
- Objectives
  - Right solution
  - Right time
  - Team & Stakeholders remain focussed on business outcome
  - Early investment returns
  - Prioritised according to business need and ability of users to accommodate change
  - No Quality compromise

DSDM Phases



Dynamic Systems Development Method

DSDM Philosophy

- Time, Cost, Quality triangle fixed early
- Contingency
- Features subjected to MoSCoW rules
    - **M**ust Have
    - **S**hould Have
    - **C**ould Have
    - **W**on’t Have this time

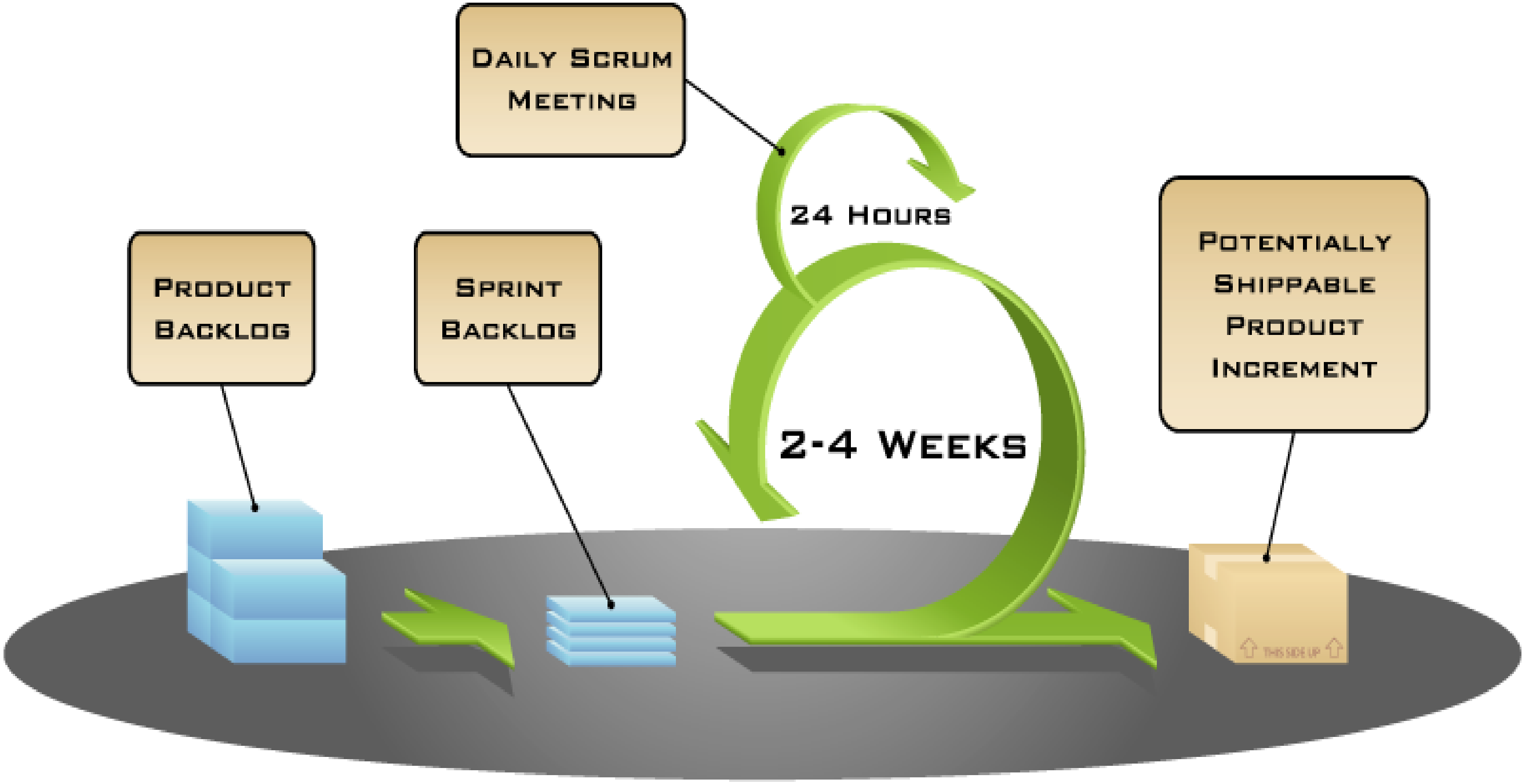
DSDM Principles

- Focus on the business need
- Deliver on time
- Collaborate
- Never compromise quality
- Build incrementally from firm foundations
- Develop iteratively
- Communicate continuously and clearly
- Demonstrate control

SCRUM

- Follows Nonaka & Takeuchi's product design principles
- "a framework within which people can address complex problems, while productively and creatively delivering products of the highest possible value"
- Uses the Experiential Learning Model
    - Sprint planning
    - Daily Scrum (Daily Plan; Daily Sprint; Daily Review)
    - Sprint Review
    - Sprint Retrospective
- www.scrum.org

SCRUM


Daily Scrum Meeting
24 Hours
Product Backlog
Sprint Backlog
2-4 Weeks
Potentially Shippable Product Increment
THIS SIDE UP

SCRUM Principles

- Transparency
- Inspection
- Adaptation
- Teamwork
  - Product owner
  - Self organized
  - No Titles
  - Shared product ownership*
  - Recognised skill-sets
- Scrum Master (servant/leader) buffer to the organisation

*but recognised individual skills such as technical developer, tester, customer

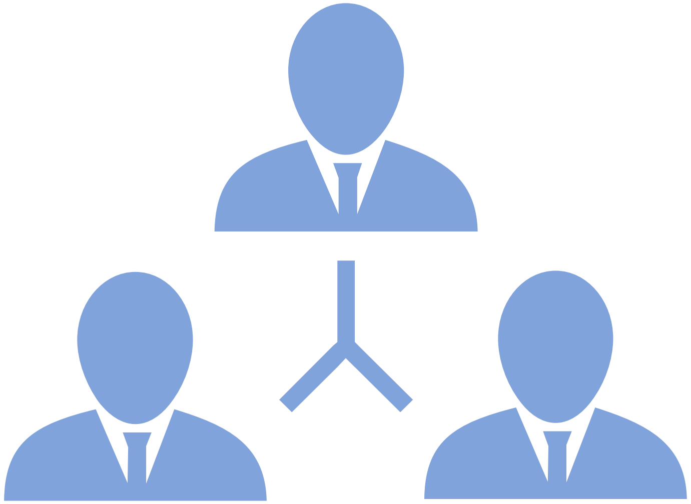
Roles in Traditional Development

business analyst key worker in design team who analyses main business aspects of the system and designs the new business processes and policies

project manager lead team member who does planning, reporting, scheduling and staff management and acts as primary point of contact for project

systems analyst identifies how technology can improve business processes and designs new business processes and the new system

technical architect technical expert who designs how the various systems components fit together to provide a stable working system

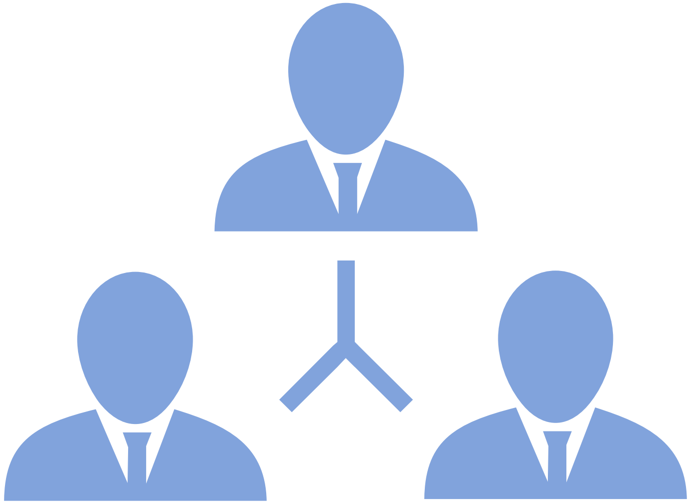


Roles in Agile Development

developer key worker in Scrum team – self-organising, no hierarchy, empowered to estimate

scrum master team member who does admin, reporting, housekeeping. Coach, facilitator, monitor. Removes impediments to the team's productivity.

product owner customer stakeholder. Prioritises/ adds detail to user stories, makes decisions on behalf of the business. Accountable for ensuring output is valuable.



Requirements / User Stories

Inception: customer works with business analyst and technical consultant to specify requirements / features in the form of user stories

User stories simple form of requirement documentation: “As a <*type of user*>, I want to be able to <*perform some interaction with the system*> in order to <*real business value/user benefit*>”.

I - Independent. The user story should be self-contained, so that there is no direct dependency on another user story.
N - Negotiable. Until part of a sprint, a story can always be changed and rewritten.
V - Valuable. A user story must deliver value to the end user.
E - Estimable. A user story must be sufficiently defined to be estimable.
S - Sized appropriately. User stories should be reasonably sized in order to prioritize & estimate with certainty. Break bigger ‘epics’ into smaller stories.
T - Testable. A user story must provide the necessary information to verify that its work can be accepted by the product owner.

Interactions

- In the Sprint, developers take a user story for coding.   ‘Done’ when feature is complete, tested and available in the continuous integration environment.

- Developer takes another user story from the sprint backlog. Progress tracked by updating burn-down charts.

- No explicit set of software engineering practices, though Extreme Programming (XP) / Test-Driven Development (TDD)/Pair programming often used.

- User testing carried out frequently to steer design and implementation decisions. Usually a free-form process where users are observed using the system.

Interactions

- In the Sprint, developers take a user story for coding. ‘Done’ when feature is complete, tested and available in the continuous integration environment.
- Developer takes another user story from the sprint backlog. Progress tracked by updating burn-down charts.
- No explicit set of software engineering practices, though Extreme Programming (XP) / Test-Driven Development (TDD)/Pair programming often used.
- User testing carried out frequently to steer design and implementation decisions. Usually a free-form process where users are observed using the system.

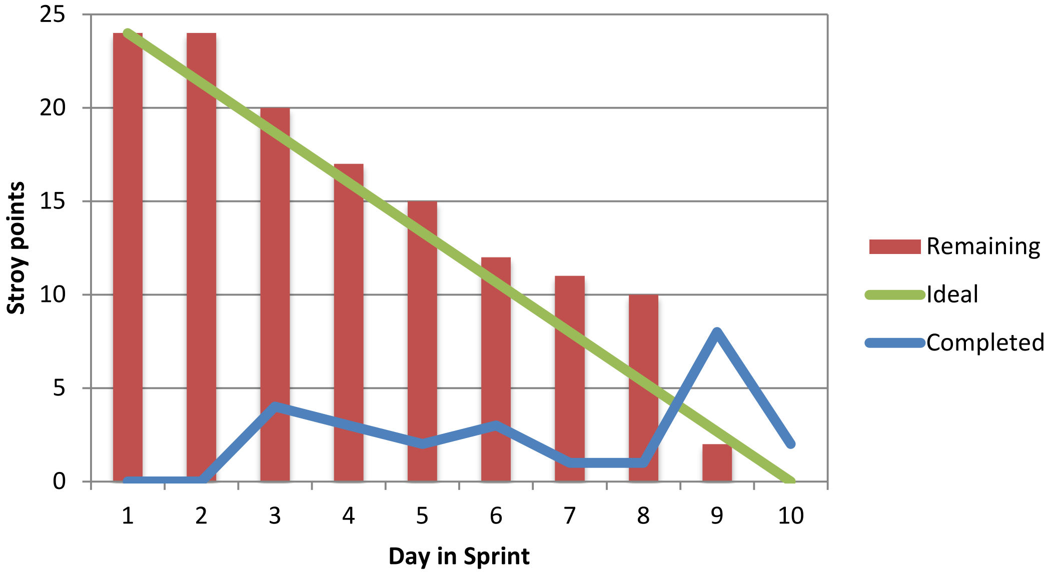
Definition of Done

code has been written
tests have been written to verify the code
code has been merged with continuous integration
all tests passed

Progress Measurement And Tracking

velocity: number of story points that have been completed in a sprint

Day	1	2	3	4	5	6	7	8	9	10
Completed	0	0	4	3	2	3	1	1	8	2
Remaining	24	24	20	17	15	12	11	10	2	0



Agile Project Activities

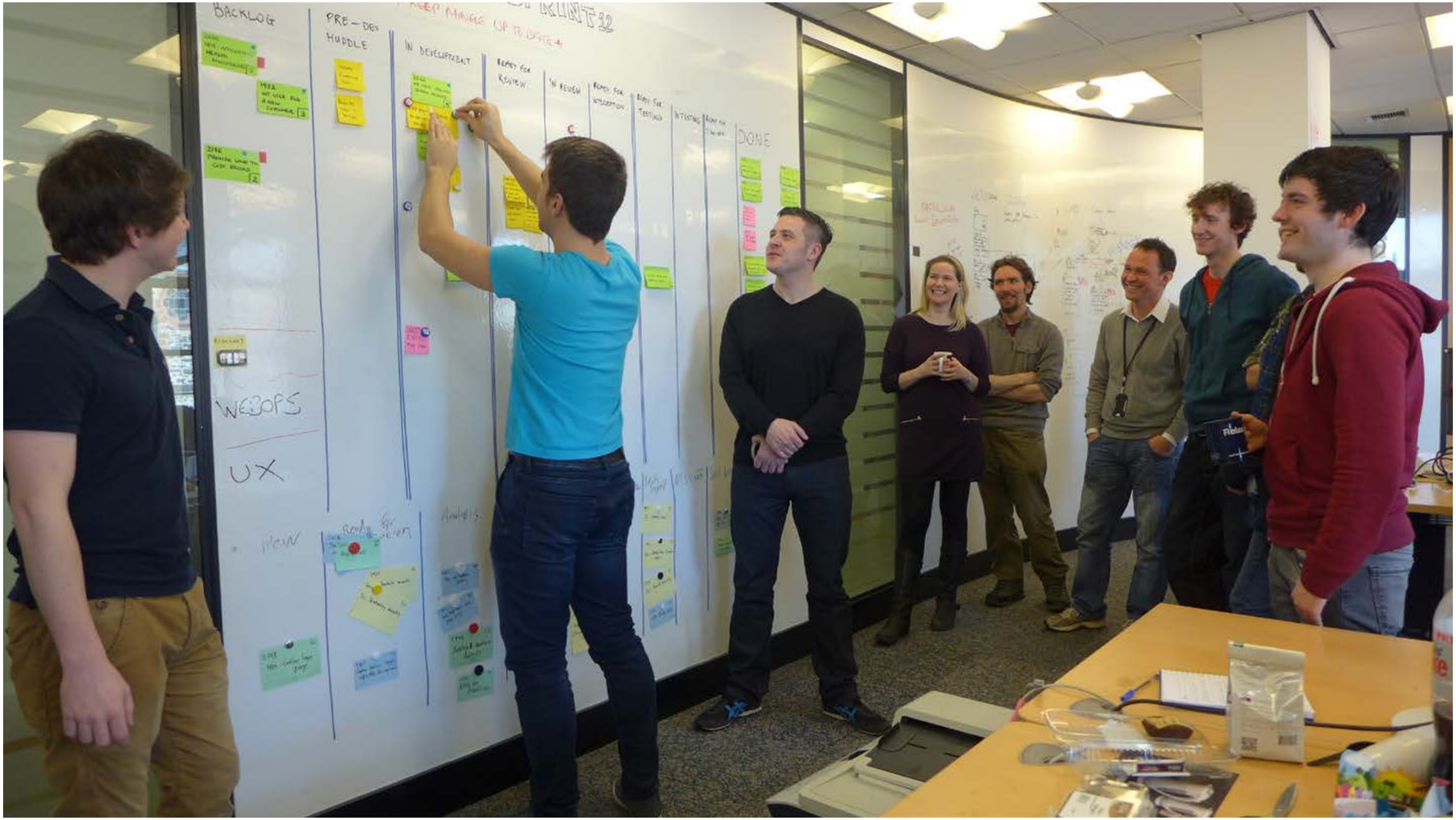
daily stand-up/scrum scrum master, developers, product owner discuss status and plan for day

show and tell team members and invited customers meet to look at output

sprint retrospective scrum master, developers, product owner discuss sprint and assess lessons learnt

project retrospective team members and invited customers meet to discuss project and assess lessons learnt

Daily Scrum

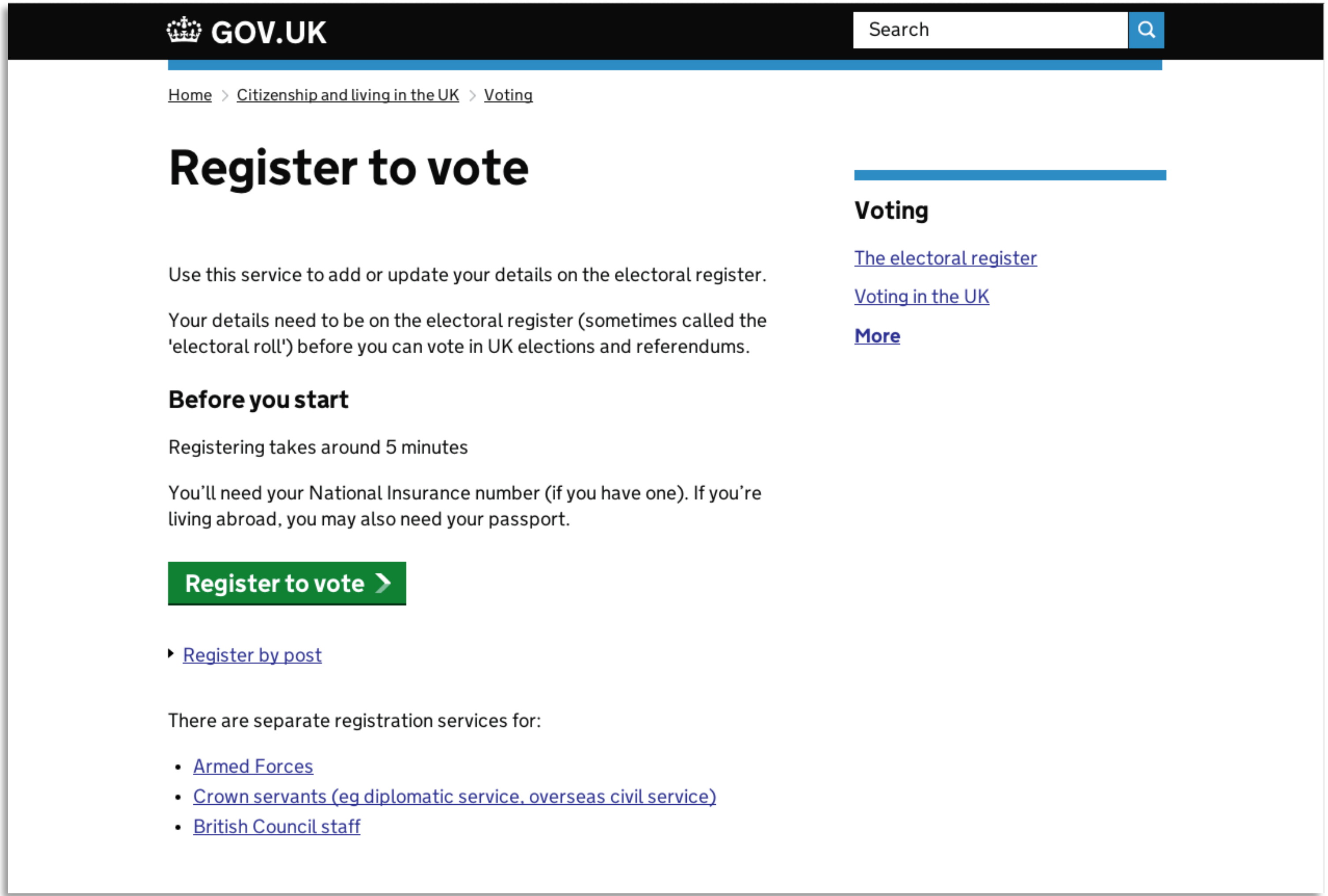

BACKLOG
PRE-DEV HUDDLE
KEEP MINGLE UP TO DATE
SPRINT 12
IN DEVELOPMENT
READY FOR REVIEW
IN REVIEW
READY FOR INTEGRATION
READY FOR TESTING
IN TESTING
DONE
WEBOPS
UX
NEW
ANALYSIS

Show and Tell

Agile Case Study

http://www.theguardian.com/technology/video/2013/jun/13/geeks-opened-up-government-video

Agile Case Study



Register to vote

Use this service to add or update your details on the electoral register.

Your details need to be on the electoral register (sometimes called the 'electoral roll') before you can vote in UK elections and referendums.

Before you start

Registering takes around 5 minutes

You'll need your National Insurance number (if you have one). If you're living abroad, you may also need your passport.

- Register by post

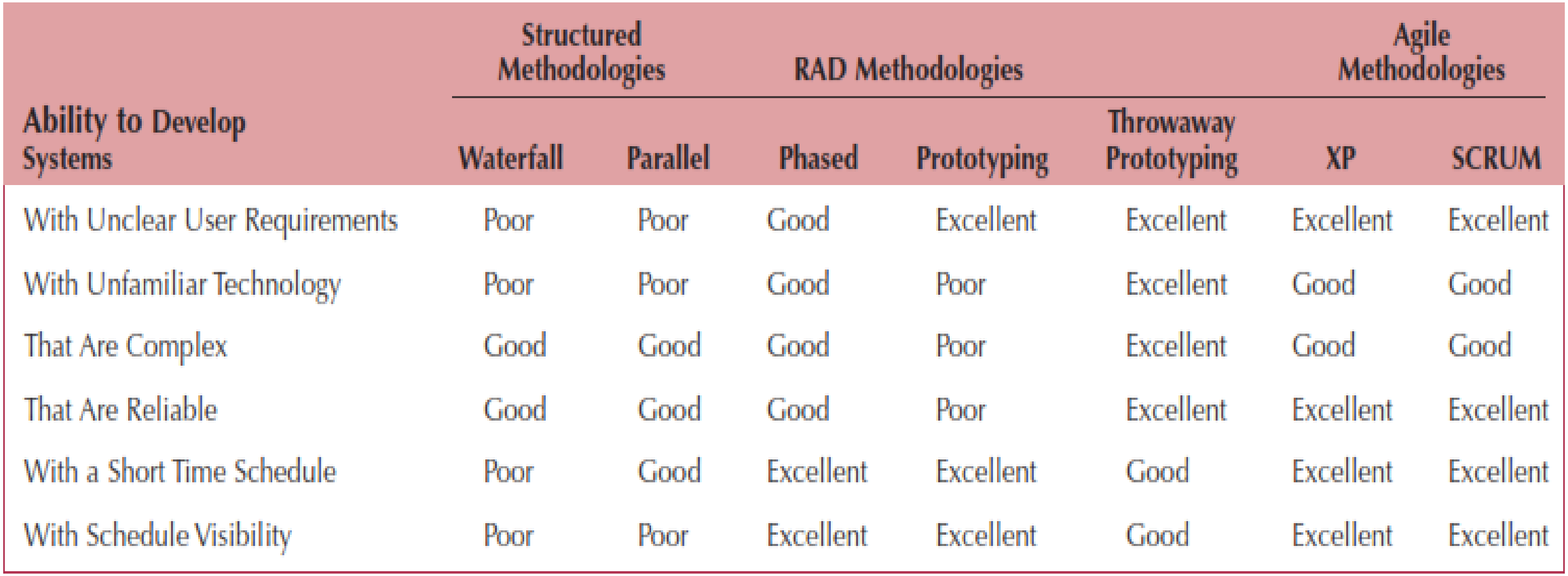
There are separate registration services for:

- Armed Forces
- Crown servants (eg diplomatic service, overseas civil service)
- British Council staff

Agile Criticisms

- Can’t sprint forever
- Co-location and distributed development
- Cost
- Endless development
- DevOps

Which Methodology to Use?



Ability to Develop Systems	Structured Methodologies		RAD Methodologies			Agile Methodologies	
	Waterfall	Parallel	Phased	Prototyping	Throwaway Prototyping	XP	SCRUM
With Unclear User Requirements	Poor	Poor	Good	Excellent	Excellent	Excellent	Excellent
With Unfamiliar Technology	Poor	Poor	Good	Poor	Excellent	Good	Good
That Are Complex	Good	Good	Good	Poor	Excellent	Good	Good
That Are Reliable	Good	Good	Good	Poor	Excellent	Excellent	Excellent
With a Short Time Schedule	Poor	Good	Excellent	Excellent	Good	Excellent	Excellent
With Schedule Visibility	Poor	Poor	Excellent	Excellent	Good	Excellent	Excellent

© Tegarden, Dennis, & Wixom, *Systems Analysis and Design with UML, 4th Edition*

Reading

- Carroll, J., Agile Project Management, In Easy Steps Ltd., Coventry, 2012
- Dennis, Wixom & Tegarden, *Systems Analysis & Design with UML*, 4th Ed, Wiley International, Chapters 3 & 4, 2012
- Highsmith, J., Agile Project Management, Pearson Ed. Boston, Mass., 2010
- M Hammer, J Champy *Reeingineering the Corporation* : HarperBusiness, 1993
- Venkatraman, N.: IT-induced Business Reconfiguration, in: Scott Morton, M. S. (Ed.): The Corporation of the 1990s, Oxford University Press, New York, 1991.

Dr. BRIAN GANNON

E:	b.gannon@bbk.ac.uk
E:	b.gannon@kainos.com
M:	07870 153855
LI:	http://lnkd.in/QDXaH6
Tw:	@bjgann