Applying Agile Methods

Question: Why doesn't everyone do it?
Question: Is it perfect?

1

Agile method applicability

- Product development where a software company is developing a small or medium-sized product for sale.
- Custom system development within an organization, where there is a clear commitment from the customer to become involved in the development process and where there are not a lot of external rules and regulations that affect the software.
- Because of their focus on small, tightly-integrated teams, there are problems in scaling agile methods to large systems.

2

2

Problems with agile methods

- It can be difficult to keep the interest of customers who are involved in the process.
- Team members may be unsuited to the intense involvement that characterises agile methods.
- Prioritising changes can be difficult where there are multiple stakeholders.
- Maintaining simplicity requires extra work.
- Contracts may be a problem as with other approaches to iterative development.

3

3

Agile methods and software maintenance

- Organizations may spend more on maintaining existing software than they do on new software development. So, if agile methods are to be successful, they have to support maintenance as well as original development.
- Two key issues:
  - Are systems that are developed using an agile approach maintainable, given the emphasis in the development process of minimizing formal documentation?
  - Can agile methods be used effectively for evolving a system in response to customer change requests?
- Problems may arise if original development team cannot be maintained.

4

4

Plan-driven and agile development

- Plan-driven development
  - A plan-driven approach to software engineering is based around separate development stages with the outputs to be produced at each of these stages planned in advance.
  - Not necessarily waterfall model – plan-driven, incremental development is possible
  - Iteration occurs within activities.
- Agile development
  - Specification, design, implementation and testing are inter-leaved and the outputs from the development process are decided through a process of negotiation during the software development process.

5

5

"Plan-driven" and "agile" development

- Is this terminology unfortunate?
- Would "traditional" and "agile" be more appropriate?

6

6

Technical, human, organizational issues

- Most projects include elements of plan-driven and agile processes. Deciding on the balance depends on:
  - Is it important to have a very detailed specification and design before moving to implementation? If so, you probably need to use a plan-driven approach.
  - Is an incremental delivery strategy, where you deliver the software to customers and get rapid feedback from them, realistic? If so, consider using agile methods.
  - How large is the system that is being developed? Agile methods are most effective when the system can be developed with a small co-located team who can communicate informally. This may not be possible for large systems that require larger development teams so a plan-driven approach may have to be used.

7

7

Technical, human, organizational issues

- What type of system is being developed?
  - Plan-driven approaches may be required for systems that require a lot of analysis before implementation (e.g. real-time system with complex timing requirements).
- What is the expected system lifetime?
  - Long-lifetime systems may require more design documentation to communicate the original intentions of the system developers to the support team.
- What technologies are available to support system development?
  - Agile methods rely on good tools to keep track of an evolving design
- How is the development team organized?
  - If the development team is distributed or if part of the development is being outsourced, then you may need to develop design documents to communicate across the development teams.

8

8

Technical, human, organizational issues

- Are there cultural or organizational issues that may affect the system development?
  - Traditional engineering organizations have a culture of plan-based development, as this is the norm in engineering.
- How good are the designers and programmers in the development team?
  - It is sometimes argued that agile methods require higher skill levels than plan-based approaches in which programmers simply translate a detailed design into code
- Is the system subject to external regulation?
  - If a system has to be approved by an external regulator (e.g. the FAA approve software that is critical to the operation of an aircraft) then you will probably be required to produce detailed documentation as part of the system safety case.

9

9

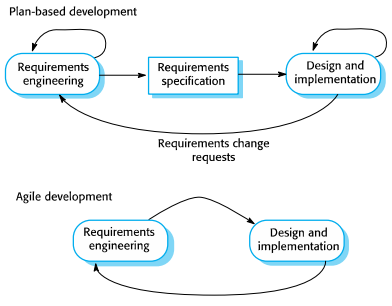
Agile methods for large systems

- Large systems are usually collections of separate, communicating systems, where separate teams develop each system.
- Frequently, these teams are working in different places, sometimes in different time zones.
- Large systems are 'brownfield systems', that is they include and interact with a number of existing systems.
- Many of the system requirements are concerned with this interaction and so don't really lend themselves to flexibility and incremental development.
- Where several systems are integrated to create a system, a significant fraction of the development is concerned with system configuration rather than original code development.

10

10

Plan-driven and agile specification



11

11

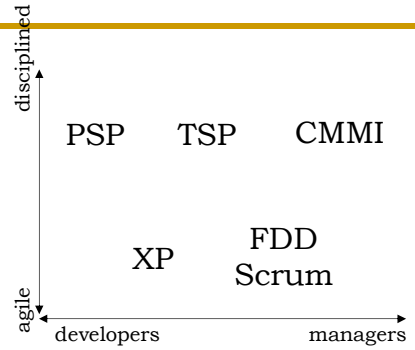
Agile Methods and Process Maturity

- **Different Goals**
  - Plan Based (Disciplined Methods)
    - Consistency
    - Stability
  - Agile Methods
    - Respond to rapid change
    - Promotes innovation
    - "WYSIWYG"

12

12

Two Dimensions



13

"Agile" and "disciplined" development

- Is this terminology unfortunate?
- Would "traditional" and "agile" be more appropriate?

14

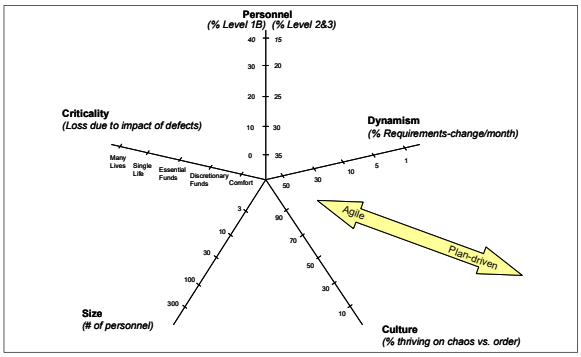
14

Matching Process to Project

- **Requirements**
  - how well understood?
  - how stable or likely to change?
  - external requirements? (DoD, FDA, etc)
- **Activities**
  - how repeatable?
    - continuous process vs. one-off project
- **Culture**
  - large or small, startup or monolith
  - group communication & experience
- **People**
  - experiences & abilities
  - personalities & priorities

15

Five Critical Dimensions



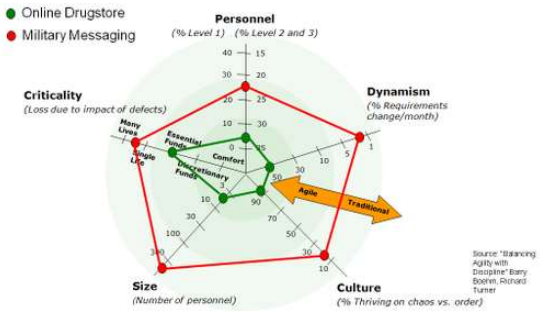
16

Labelling in Boehm Turner Model

- Is it "Requirements Change" or "Requirements Uncertainty"?
- Is it thriving on chaos versus order? Or is it simply a different paradigm? Is agile really "chaotic" or just "flexible"?

17

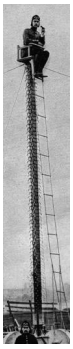
Five Critical Dimensions



18

Some Observations on Balancing

1. Neither agile nor plan-driven methods provide a silver bullet
2. Agile and plan-driven methods have home grounds where each clearly dominates
3. Future developments will need both agility and discipline
4. Some balanced methods are emerging
5. It is better to build your method up than to tailor it down
6. Methods are important, but potential silver bullets are more likely to be found in areas dealing with
   - People
   - Values
   - Communications
   - Expectations management



19

1. No Silver Bullet

- Brooks's werewolf concerns
  - Complexity, conformity, changeability, invisibility
- Agile methods
  - On target for changeability and invisibility
  - Miss on complexity and conformity
- Plan-driven methods
  - On target for conformity and invisibility
  - Miss on complexity(?) and changeability
- Bullets can lose their efficacy as "wolves" evolve

20

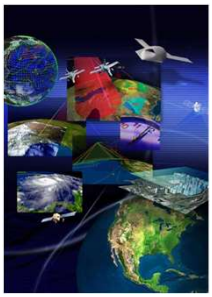
2. Home Grounds Exist

- Agile and plan-driven methods have definite home grounds
  - Environment where they are most likely to succeed
  - Extremes are rarely populated
- Five dimensions can help illustrate a project's or organization's home ground relationship
  - Size, Criticality, Dynamism, Personnel, Culture

21

3. Future Applications Need Both

- Historically
  - Many small, non-critical, well-skilled, agile culture, rapidly evolving projects
  - Many large, critical, mixed-skill, ordered culture, stable projects
- In the future?
  - Large projects are no longer stable
  - Maintenance of extensive process and product plans will become too expensive
  - Complexity and conformity werewolves are waiting for agile projects
  - Attributes of both approaches will be needed
  - Hardware delivery/provisioning models greatly affect software architectures and associated software development approaches?



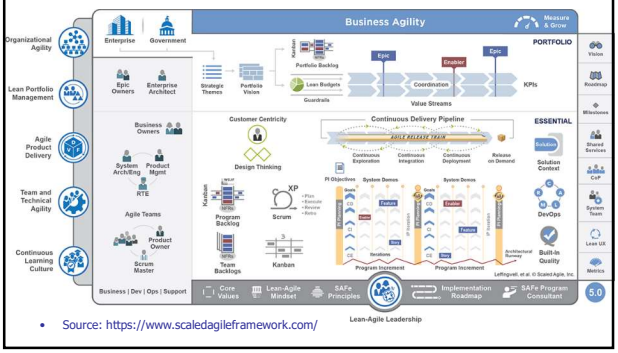
22

4. Balanced Methods have Emerged

- Agile methods
  - Crystal Orange
  - DSDM
  - FDD
  - Lean Development
  - SAFe
- Plan-Driven methods
  - Rational Unified Process
  - CMMI
- Hybrid
  - Boehm-Turner Risk-based
  - Manzo (AgileTek) Code Science/Agile Plus
  - SAFe (Scaled Agile Framework)

23

SAFe



- Source: https://www.scaledagileframework.com/

24

5. Build up – Don't Tailor Down

- Plan-driven methods traditionally
  - Have over-defined processes?
  - Advocate (or require) tailoring
  - Are rarely tailored well / easily
- Agilists traditionally
  - Begin with the minimum
  - Add as needed (and justified by cost-benefit)
  - Have multiple core sets

25

6. Methods aren't always the answer

- Agile movement has echoed a long line of warning calls
- Success of agile may be due as much to people factors as to technology
- Valuing people over processes is the most important factor in the agile manifesto?

I know I saw something about that in the process somewhere…

26

Conclusions

- Plan-driven and agile methods both aim to
  - Satisfy customers
  - Meet cost and schedule parameters
- Home grounds exist, but the opportunity for integration is expanding
- CMMI supports balancing methods
  - Flexible application of the model allows both plan-driven and agile methods
  - PA support to risk-based balancing process
- For more on the risk-based process, see
  - Boehm/Turner, Balancing Agility and Discipline: A Guide for the Perplexed, Addison Wesley, Boston, 2003.

27

Philosophical in-class discussion…

- A question of discipline - Are agile methods more/less disciplined than traditional approaches?
- Is the evolution of software development approaches sometimes like the seasonal / annual procession of dress fashion?
- Which approach to software development is the best approach?
- Does agile software development require *premium* software developers?
- How should we think in relation to the dogma and exclusivity projected by proponents from all approaches, traditional and agile?
- Do current pay-per-use hardware innovations affect software development processes and architectures?
- Is there a problem with the inconsistent/incorrect use of language?
  - Are agile approaches not also *planning* oriented?
  - Can traditional approaches not offer *flexibility* in relation to requirements?
  - What' s *lightweight* about Scrum / XP?

28