

Compliant Agile Development

Date: 21 Oct 2009

Version: Final

Authors: Christopher Cieslok & Ian Francis

Compliant Agile Development

1 Life Science Development Models

2 Adaptive Models

3 Challenges to agile adoption

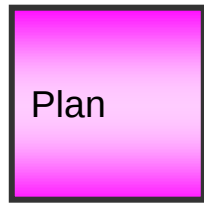
4 Compliant Agile Overview

5 Limitations

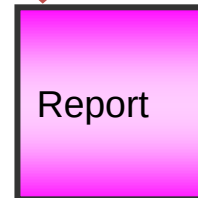
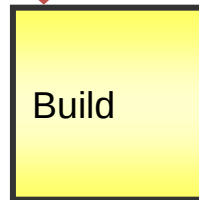
6 Compatibility Assessment



LIFE SCIENCES DEVELOPMENT MODELS



Standard approaches to software development
in Life Sciences are “Predictive”



Waterfall



LIFE SCIENCES DEVELOPMENT MODELS

Characteristics of Predictive Approaches

- Rely on users knowing what they want early in process
- Sequential phases, usually with a little overlap
- Changes later in the process are discouraged
- Software for User Acceptance delivered at end of process (after months...years?)
- User collaboration during requirements gathering (beginning) and acceptance testing (end)

- Predictive approaches are not “bad”...but are often badly applied e.g.
 - requirements being defined in isolation from the system and then...
 - being difficult to change early in the life cycle which results in...
 - delivering a solution that does not meet user's real needs...
 - causing dissatisfaction in the users
 - does not provide a return on investment



ADAPTIVE MODELS

Characteristics of Adaptive (Agile) Approaches

- Allows users to evolve ideas of what they want throughout the process
- Continuous specification/test for each requirement (user story)
- Software for User Acceptance is delivered frequently (weeks rather than months)
- Close, daily cooperation between business people and developers

- Adaptive approaches are useful...but have concerns in regulated industries
 - Often deliver **inadequate documentation**
 - Lack of **defined scope**
 - **Non-functional quality attributes** are not easy to define as user stories
 - Hard to assess **realistic estimates of work effort** needed, because unknown scope/requirements at the beginning of the project



CHALLENGES TO AGILE ADOPTION

Scope Creep

- Scope should be clearly defined, for cost control and time/resource management
- Too often this means the scope is locked early and at a detailed level
- But we should identify a difference between "scope creep" and "functional evolution"
- Define scope at a high level with a Business Solution Concept
- Detailed specifications open to evolution and (managed) change

Software Control

- Key to any well managed development
- Agile type developments are no different
- Efficient configuration, change and release management processes applied
 - Good Engineering Practices (GEP)
 - Use appropriate tool sets (e.g. Subversion)
 - Ensure distinction between development, QA and production releases



CHALLENGES TO AGILE ADOPTION CONT.

Documentation

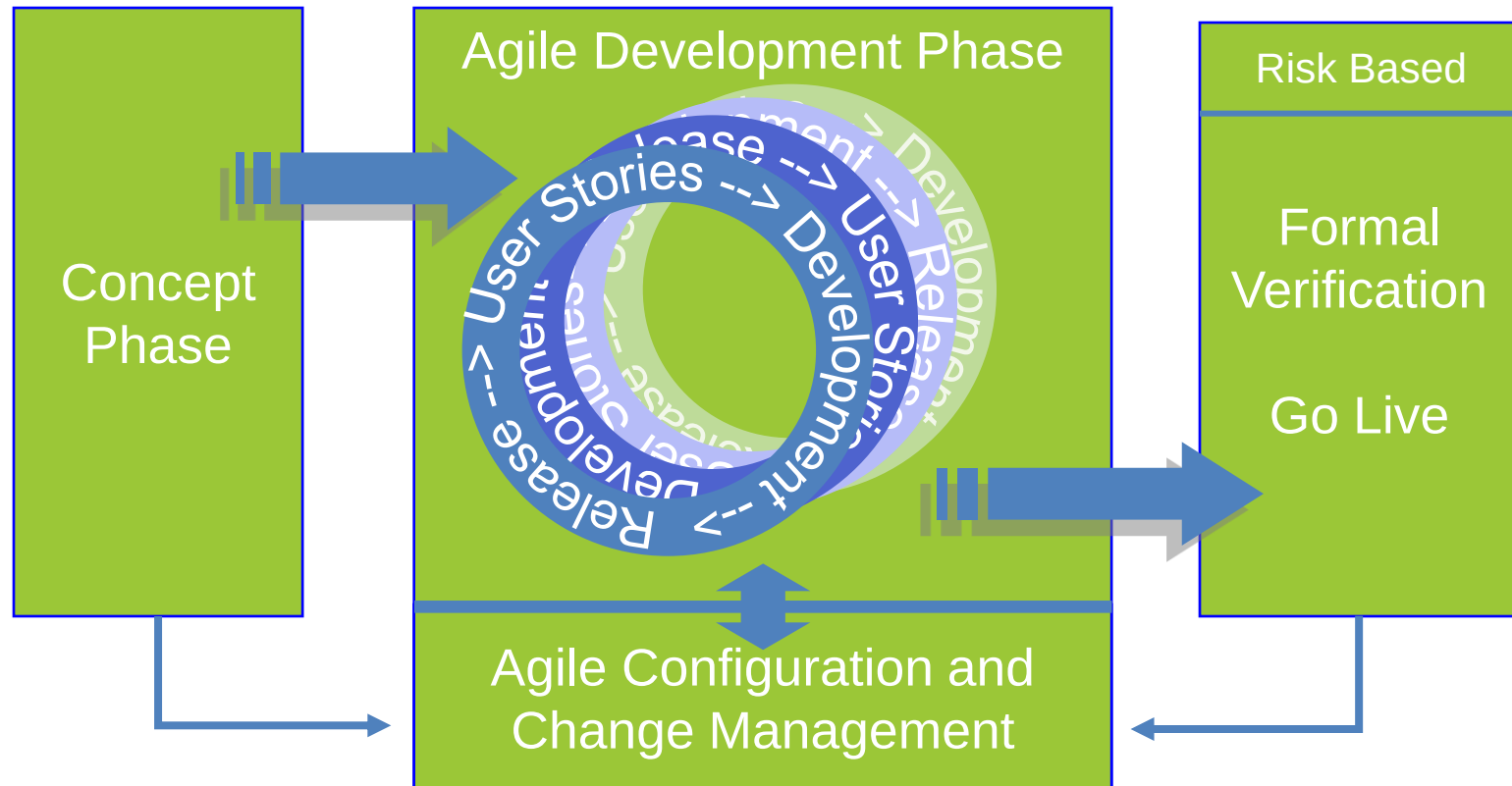
- Need to have the process for providing quality documentation **built in** to the Agile process
- Combining and scaling documents where appropriate

Technical Foundation

- Software development has a set of prerequisite foundations
- These can include
 - larger application frameworks (e.g. SAP, SAS)
 - programming languages,
 - compilers,
 - operating systems,
 - hardware,
 - networks,
 - databases
- These should be appropriately qualified **before** beginning an Agile development



COMPLIANT AGILE OVERVIEW



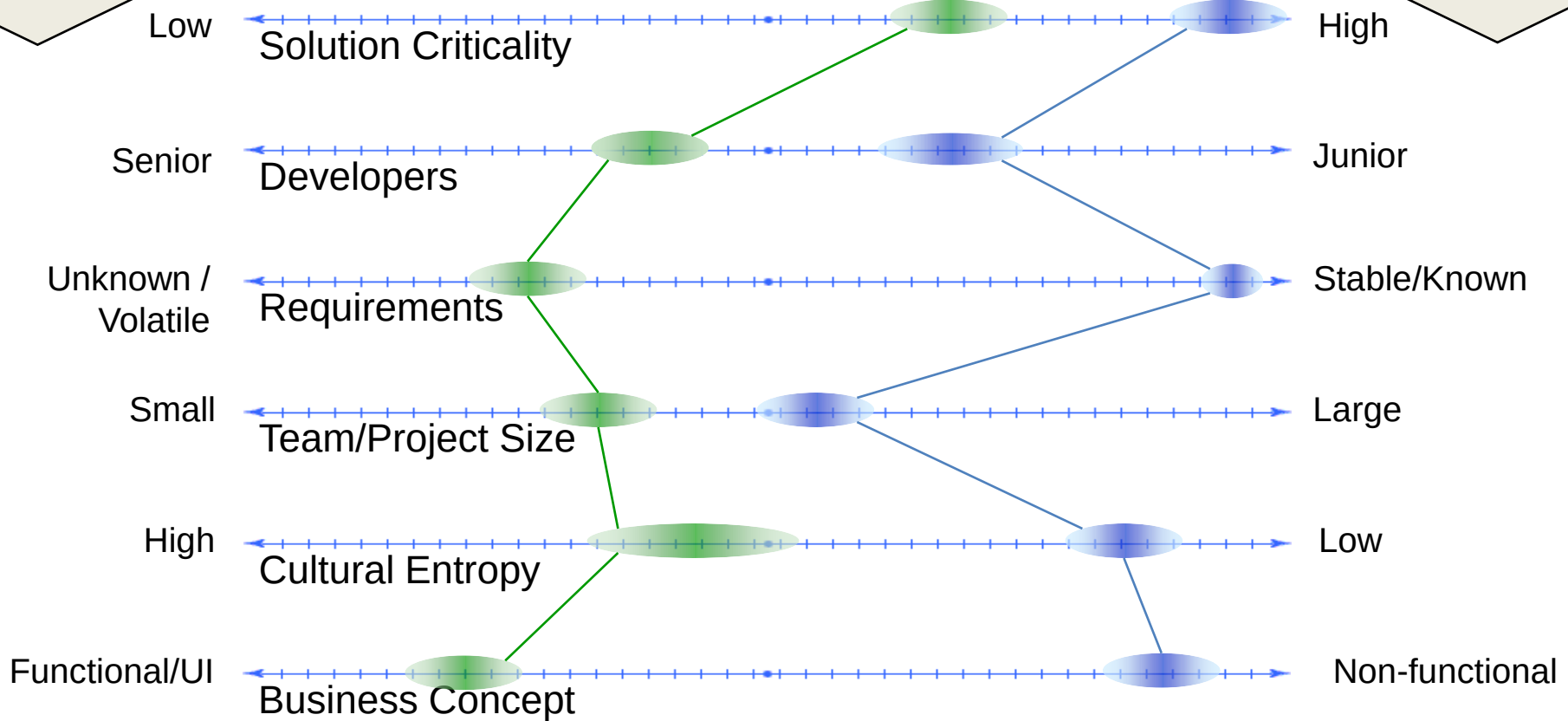
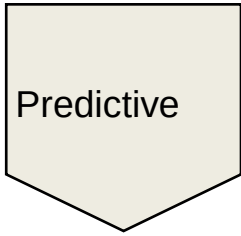
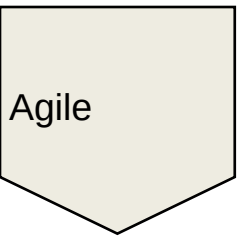


LIMITATIONS

- We recognise that an Agile approach is not applicable to all systems, projects or even components of projects
- As part of an overall risk management process, we recommend that the suitability of Agile approach is assessed
- Assessment can be applied at appropriate levels
 - the entire solution
 - individual work streams
 - functional areas



AGILE COMPATIBILITY ASSESSMENT



Example: Web interface to regulated system could indicate that an Agile approach is justified

Example: Access controls for regulated system indicate that predictive approach is justified



THANK YOU VERY MUCH

...any questions?



CONTACT

Christopher Cieslok

○ christopher@cieslok.net

Ian Francis

○ ian.francis@businessdecision.com