Heavy vs Light Methodologies: Bulimic or Anorexic?

Fernando Brito e Abreu (fba@di.fct.unl.pt)

Universidade Nova de Lisboa (http://www.unl.pt)

QUASAR Research Group (http://ctp.di.fct.unl.pt/QUASAR)

Abstract

- *From anorexic to bulimic*
- *Overview of heavy-weight methodologies*
- *Origins of light-weight methodologies*
- *The "Manifesto"*
- *Agility example: XP*
- *The dark side of the light*
- *Planned (RUP) vs Agile (XP)*
- *When to be agile?*

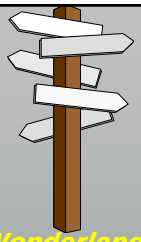
3

Some hype ...

Plan-driven methodologies	Agile software development
Heavy-weight methodologies	Light-weight methodologies
CMM, ISO9000-3, ISO12207, PSP, TSP, ISO15504 (SPICE), RUP, CMMi, ...	Extreme Programming (XP), Scrum, Feature-Driven Development (FDD), Adaptive Software Process, Crystal Light Methodologies, Dynamic Systems Development Method (DSDM), Lean Development
Fat? Bulimic?	Thin? Anorexic?

4

From anorexic to bulimic



"If you don't know where you are going, every road will take you there..."

Alice in Wonderland

- Many projects run unconstrained
  - Total freedom for artists ☺
  - Unpredictable results ☹
- DoD Mil. Std 2167A
  - Highly constrained, high organizing overheads ☹
  - Very well defined deliverables ☺

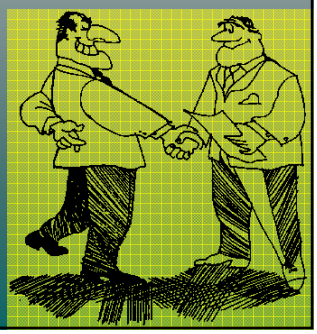
5

Overview of heavy-weight methodologies

- Processes and tools
- Comprehensive documentation
- Contract negotiation
- Following a plan

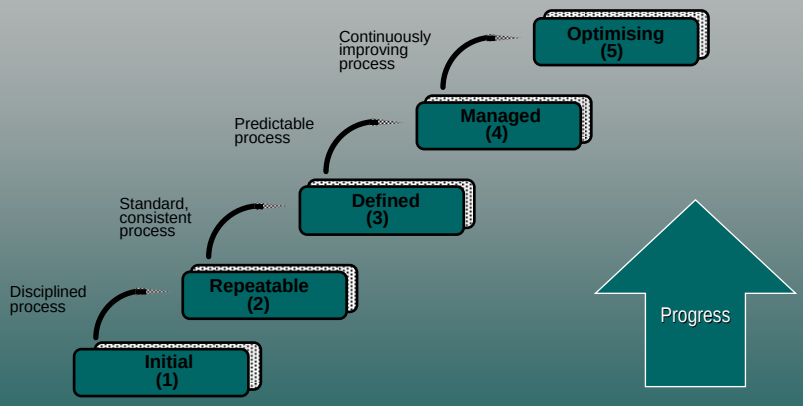
"On projects with more than 250 people, methodology will have almost no impact on success or failure – politics will dominate."

Jim Highsmith

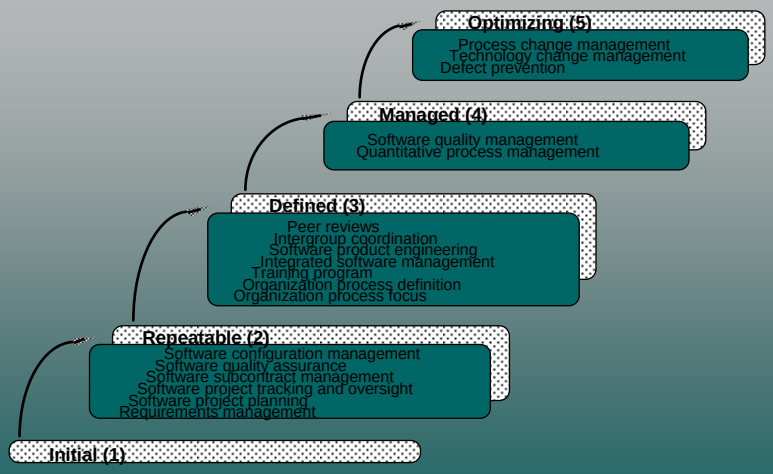


6

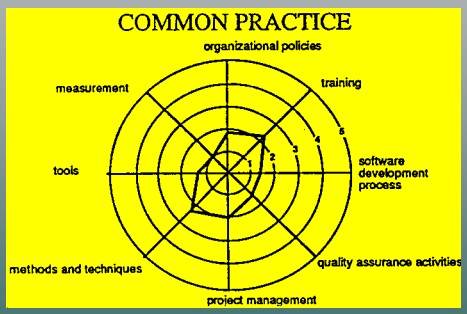
CMM - Capability Maturity Model

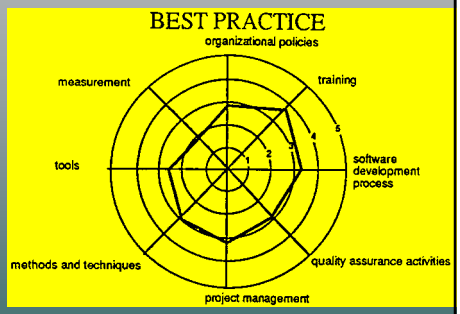


CMM - Capability Maturity Model



Process Advisor (Roger Pressman & Associates)





ISO 15504

- SPICE - Software Process Improvement and Capability dEtermination
- Initiative of **WG10** (Process Assessment) - ISO/IEC JTC1/SC7 (Software Engineering)
- OBJECTIVES:
  - Unify software process assessment efforts
  - Elaboration of a set of international standards

ISO 15504
Organizations involved

Australian Software Quality Research Institute

Bell Canada

Northern Telecom

Bell Northern Research (BNR)

BOOTSTRAP Consortium

British Telecommunications Plc.

Centre de Recherche d'Informatique de Montréal

Defense Research Agency, UK

European Software Institute

Software Engineering Institute

Etnoteam, Italy

University of Oulu, Finland

Bellcore, EUA

... and other organizations from Japan, South Africa, France, Ireland, Spain, ...

ISO 15504

Considers 5 Generic Process Categories:

- CUS - *customer-supplier process category*
- ENG - *engineering process category*
- PRO - *project management process category*
- SUP - *support process category*
- ORG - *organization process category*

Capability Maturity Model Integrated (CMMI®)

- Is an integrated model to propel process improvements in systems engineering and software engineering.
- The model encompasses:
  - 5 maturity levels
  - 25 Process Areas (PAs)
  - Several flavors (SE/SW/IPPD/SS)

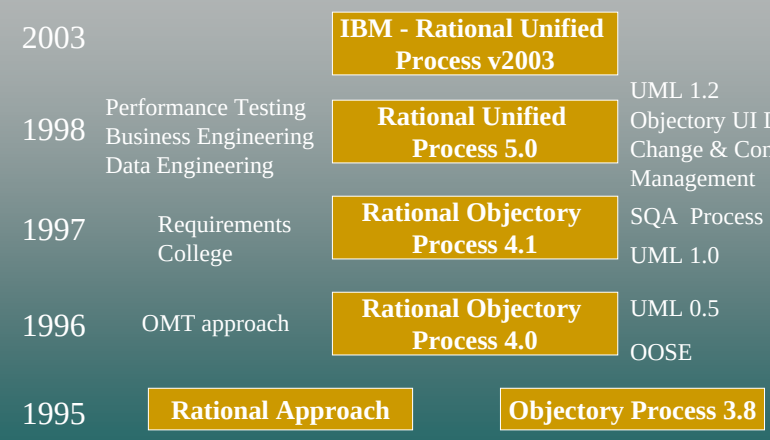
13

Capability Maturity Model Integrated (CMMI®)

- **Level 2** - Requirements Management, Project Monitoring and Control, Project Planning, Supplier Agreement Management, Configuration Management, Process & Product QA, Measurement & Analysis
- **Level 3** - Requirements Development, Technical Solution, Product Integration, Organizational Training, Verification Validation, Risk Management, Decision Analysis & Resolution, Integrated Project Management, Organizational Process Focus, Organizational Process Definition
- **Level 4** - Quantitative Project Management, Organizational Process Performance
- **Level 5** - Organizational Innovation & Deployment, Causal Analysis & Resolution
- Additional Requirements of IPPD - Changes to Integrated Project Management, Integrated Teaming and Organizational Environment for Integration
- Additional Requirements of SS - Integrated Supplier Management

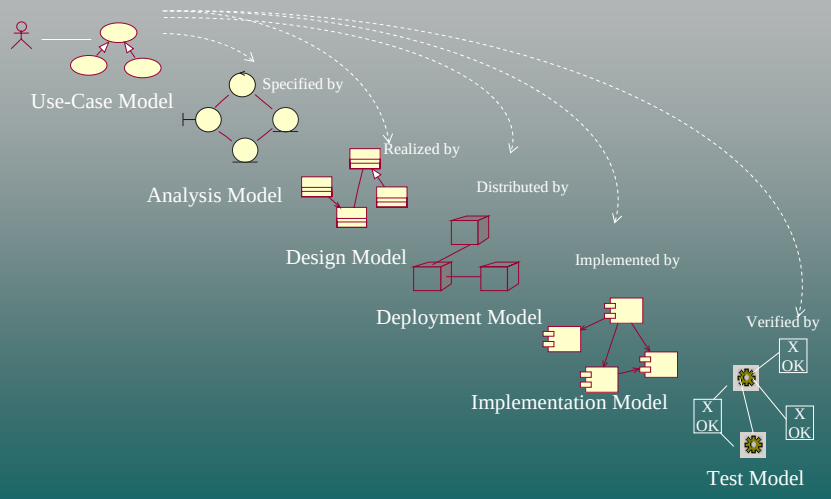
14

Rational Unified Process (RUP)



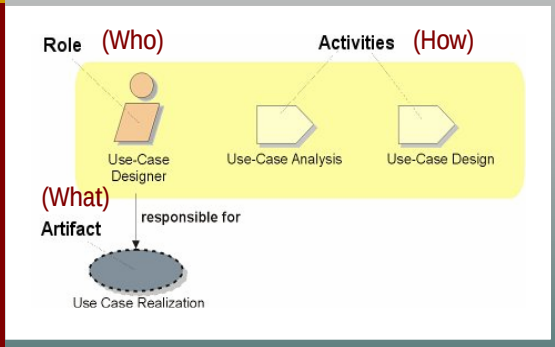
15

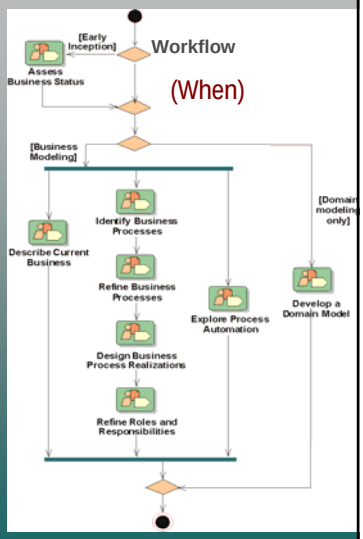
Rational Unified Process (RUP)



16

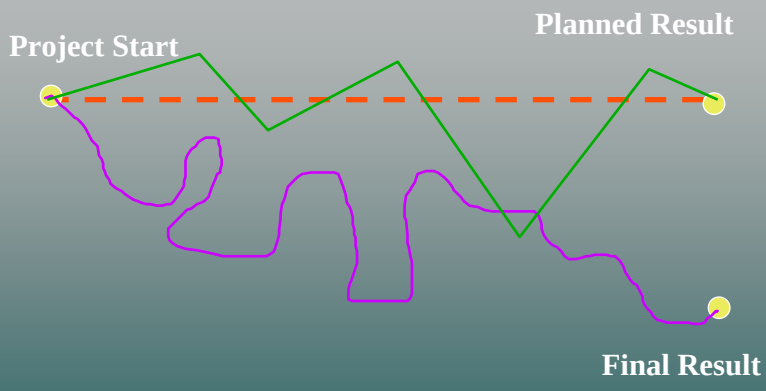
Rational Unified Process (RUP)





17

Origins of light-weight methodologies



18

Origins of light-weight methodologies

- Agility
  - The ability to both create and respond to change in order to profit in a turbulent business environment
    - Companies need to determine the amount of agility they need to be competitive
- Chaordic (ex: agile view)
  - Exhibiting properties of both ***cha***os and ***ord***er
    - The blend of chaos and order inherent in the external environment and in people themselves, argues against the prevailing wisdom about predictability and planning
    - Things get done because people adapt, not because they slavishly follow processes
  - An agile view is a chaordic view

19

The Agile Manifesto Subscribers

Alistair Cockburn
Andrew Hunt
Arie van Bennekum
Brian Marick
Dave Thomas
James Grenning
Jeff Sutherland
Jim Highsmith
Jon Kern
Ken Schwaber
Kent Beck
Martin Fowler
Mike Beedle
Robert C. Martin
Ron Jeffries
Steve Mellor
Ward Cunningham

20

The Agile Manifesto [Feb 2001]

"We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

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

That is, while there is value on the items on the right, we value the items on the left more."

21

Agile Management Issues

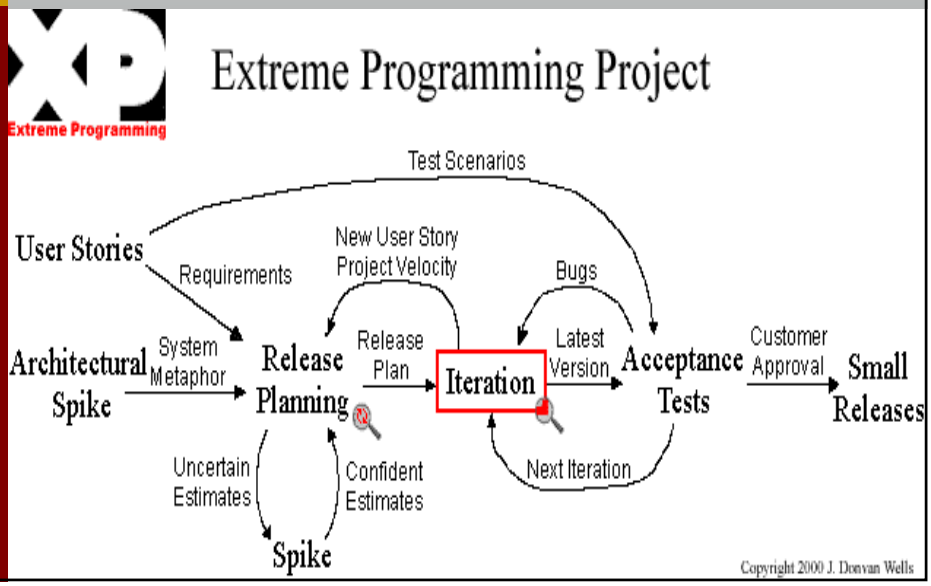
- Promote teambuilding and trust
- Set an open tone with the customer(s) organizations
- Interpret and translate risks for overall program integration and a common view
- Have a robust, flexible, and adaptable configuration and data management system

22

Summary of Agile Characteristics

- Adaptability rather than predictability
- People rather than development process
  - Being agile means accepting that outcomes are not predictable and that processes are not repeatable
- Collaborative values and principles
- A barely sufficient methodology
  - "the conventions we agree to"
  - Processes are described in manuals; practices are what happen in reality

Agility example: XP

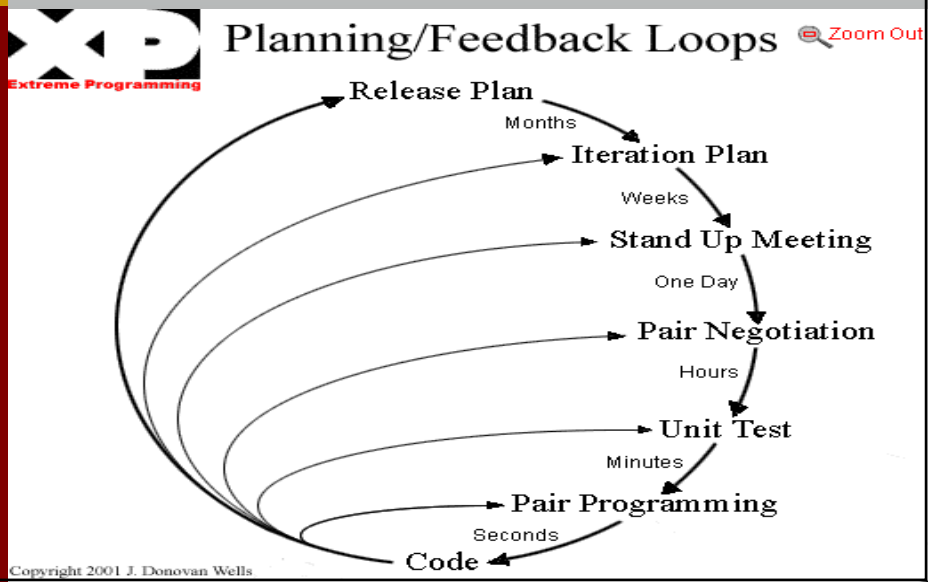


XP – Practices

- The Planning Game
- Small Releases
- System Metaphor
- Simple Design
- Testing
- Refactoring
- Pair Programming
- Collective Ownership
- Continuous Integration
- 40-hour week
- On-site-Customer
- Coding Standards

25

XP – Schedule …



26

The dark side of the light …

"XP Considered Harmful … for Reliable SW Development"

[Gerold Keefer, 2002]

- The embrace change value …
- The practice of refactoring …
- The simplicity value …
- The practice of pair programming …

…

27

Planned (RUP) vs Agile (XP)

RUP

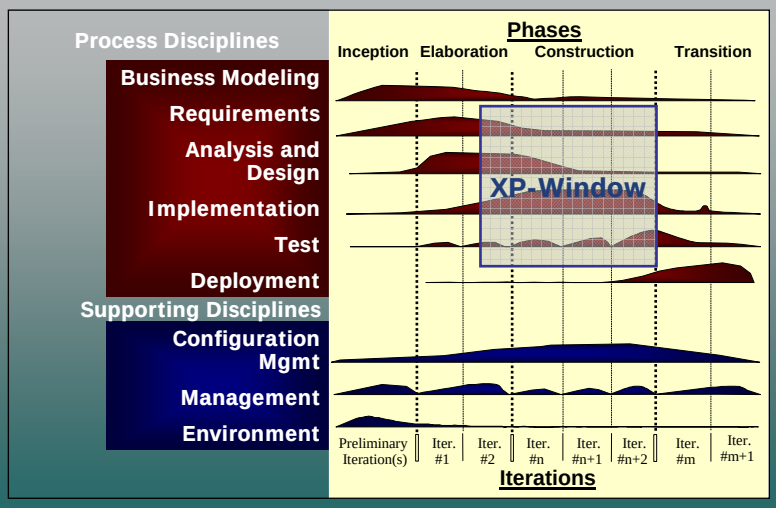
- Business Modeling
- Req. Management
- Analysis & Design
- Implementation
- Test
- Deployment
- CCM
- Project Management
- Environment

XP

- Coding
- Testing
- Listening
- Designing

28

Planned (RUP) vs Agile (XP)



29

When to be agile?

- Problems characterized by change, speed, and turbulence are best solved by agility.
  - Accelerated time schedule combined with significant risk and uncertainty that generate constant change during the project.
- Is your project more like drilling for oil or like managing a production line?
  - Oil exploration projects need Agile processes.
  - Production-line projects are often well-served by rigorous methodologies.

30

That's all folks ☺

Fernando Brito e Abreu

fba@di.fct.unl.pt

http://ctp.di.fct.unl.pt/QUASAR

- Questions?

References

Agile Software Development Ecosystems, Jim Highsmith

Agile Software Development, Alistair Cockburn, Pearson Education, Inc, Boston, MA, 2002

Agile Modeling: Effective Practices for Extreme Programming and the Unified Process, Scott Ambler

Agile Development, Rich McCabe, May 2003

Agile Software Development with Scrum, Ken Schwaber and Mike Beedle, Prentice-Hall, Inc., 2002

Extreme Programming Explained: Embrace Change, Kent Beck

Refactoring: Improving the Design of Existing Code, Martin Fowler

Adaptive Software Development, A Collaborative Approach to Managing Complex Systems, Jim Highsmith, Dorset House Publishing, New York, 2000.

A Practical Guide to Feature-Driven Development, Stephen Palmer and John Felsing

Rising, L. and N. S. Janoff (2000). The Scrum Software Development Process for Small Teams. IEEE Software. July/August 2000

Foundations of Lean Development: The Lean Development Manager's Guide, Vol 2, Robert Charette

Extreme Programming Explained, Kent Beck, Addison Wesley, 2000

The Rational Unified Process, An Introduction, Second Edition, Philippe Kruchten, Addison-Wesley, 2000

Refactoring: Improving the Design of Existing Code, Martin Fowler et al., Addison-Wesley, 2000

Extreme Programming Installed, Ron Jeffries, Ann Anderson, Addison-Wesley