



Datateknisk Forum Report

Iterative Software Development A Practical View

Otto Vinter, Robert Olesen, Morten Korsaa

DELTA, Department of Software Engineering,

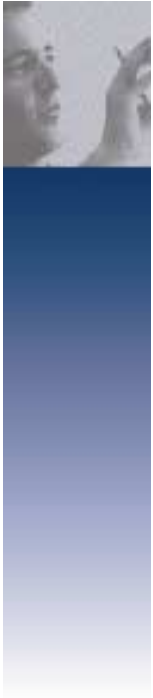
Tel: +45 7219 4000, Fax: +45 7219 4001

otv@delta.dk, ro@delta.dk, morten@korsaa.com

www.delta.dk/iterativ

T H E K N O W L E D G E C E N T R E





Content

- Background
- Overview of the Report
- Principles for Iterative Software Development
- Related aspects
- Implementation

What is it? It is a solution to a problem!

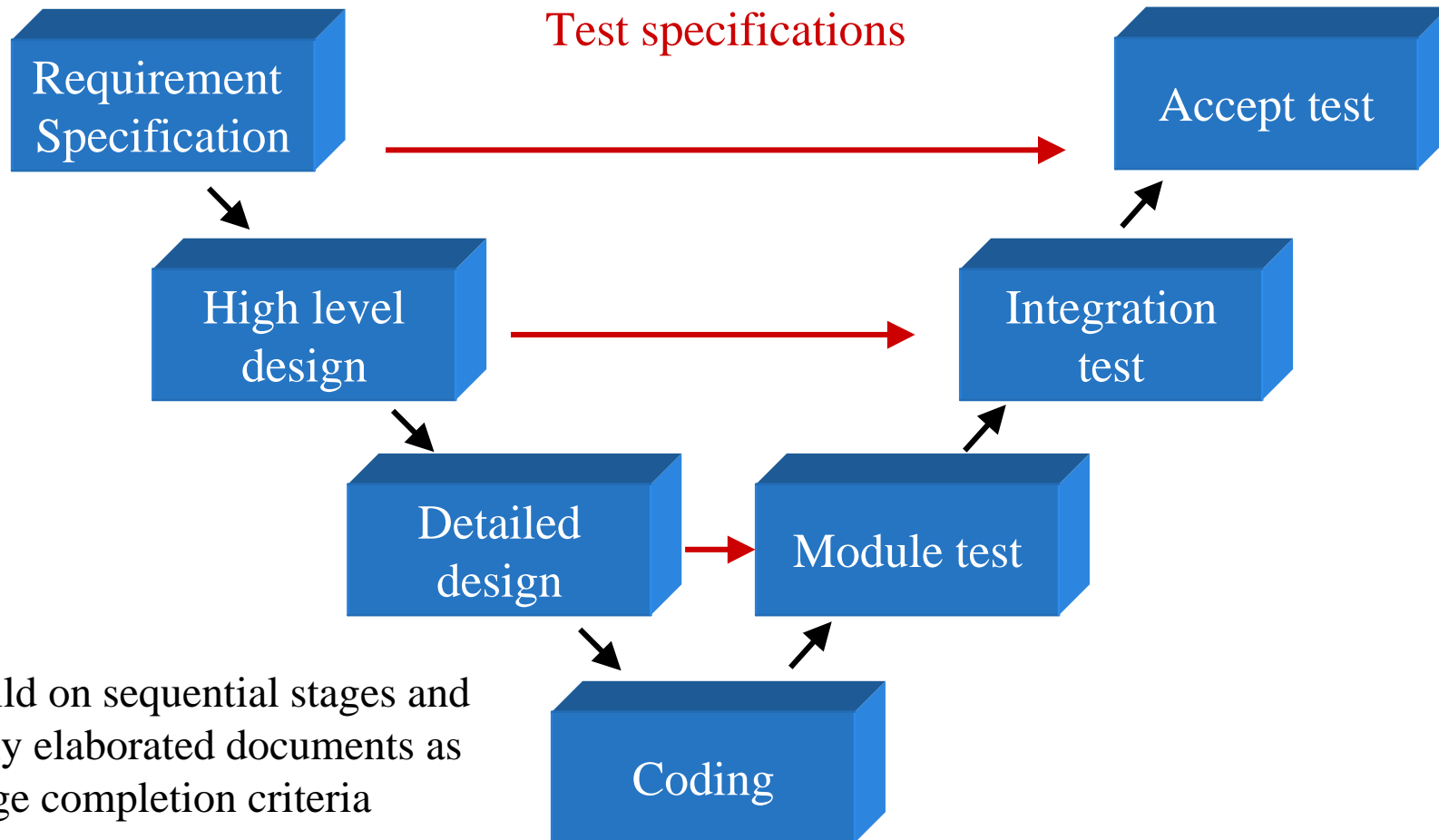
Risk:
The quality of the available information is insufficient!

Handle the risk!
Waterfall
planning, and
solving problems
as they occur.

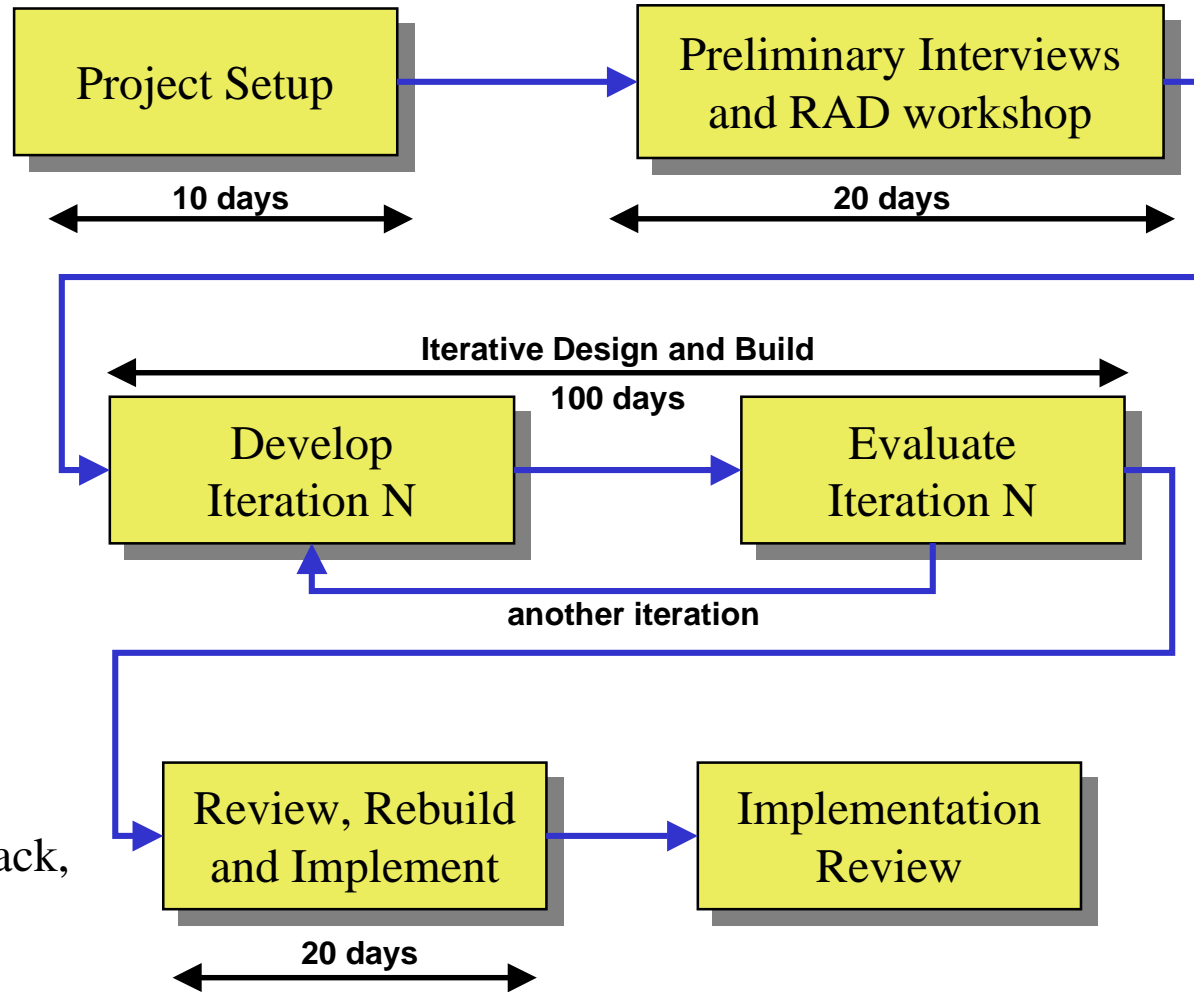


Eliminate the risk!
Iterative planning.

Waterfall Model Example - The V Model



Iterative Model Example - RAD Model



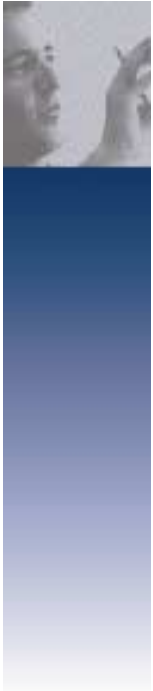
Initiate

Develop

Deploy

Build on incremental releases, user participation, frequent feedback, and change





Waterfall

Assumptions

- The customer knows what he wants
- The requirements are frozen (changes are exceptions).
- Phase reviews are used as control and feedback points

Characteristics of success

- Stable requirements
- Stable environments
- Focus is on the big picture
- One, monolithic delivery

Iterative

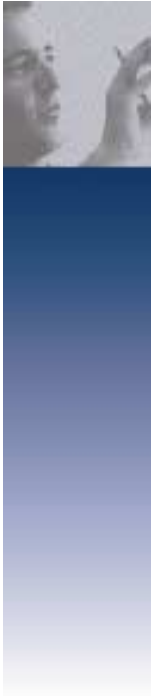
Assumptions

- The customer cannot express exactly what he wants
- The requirements will change.
- Reviews are done continuously for control and feedback

Characteristics of success

- Fast and continuous user feedback
- Floating targets
- Focus is on the most important features
- Frequent releases





Purpose of the report

- Understand and evaluate if iterative development will support your business.
- Choose among the most common iterative development methods the one that is most suitable for you.

Audience

- Primarily developers, project managers, and line managers in development.
- Secondarily method/architecture groups and process/quality departments.

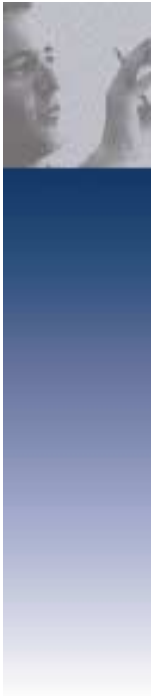




Metaphor: A Traveller's Guide

- Introduction to the area
- The history
- The people and the culture
- Tours in the area
- Description of places and sights
- Practical hints and tips
- Commonly used phrases
- References to more information

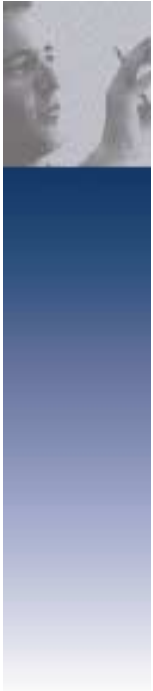




A Guide to Iterative Development

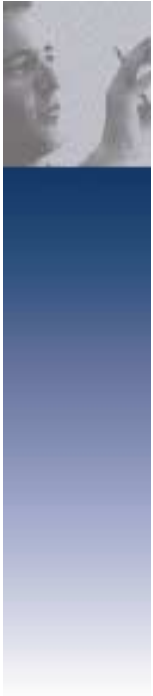
- Introduction what is it
- The history where does it come from
- Its purpose why, when, and how to use it
- Principles what are the characteristics
- Methods what frameworks exist
- Related activities what is important to know
- Experiences what have others done
- References where to find more detail





The Principles

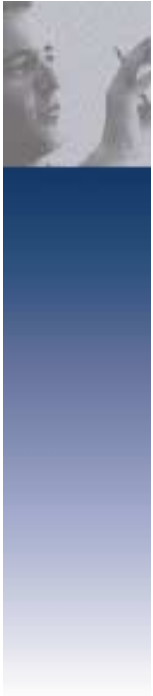
- We have derived some principles for Iterative Software Development to guide the implementation.
- They can be used to judge if a given practice is iterative. Then you can assess whether the practice is supporting your business, or it need to be changed.
- Use them literally every time you are discussing a practice. The principles may not be the final truth, but if you use them to guide a discussion, you will end with a conclusion of what is true for you. And that is the most important.



Principles of Iterative Development

- Time has priority over functionality
- Get the most bang for the buck
- Power to the customer
- Empowered teams
- Adaptability
- Short life cycles
- Fast feedback

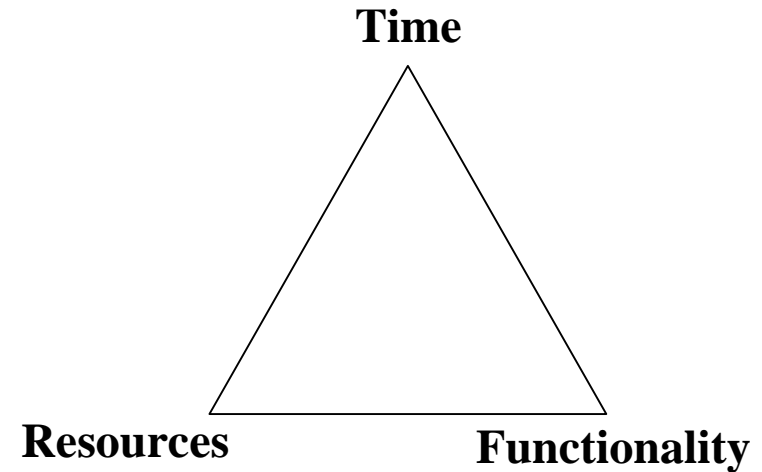




“Time has Priority over Functionality”

- Time
 - Fixed- by all means
- Resources
 - In reality fixed
- Functionality
 - The only variable

Timeboxing



You can't stop the clock!





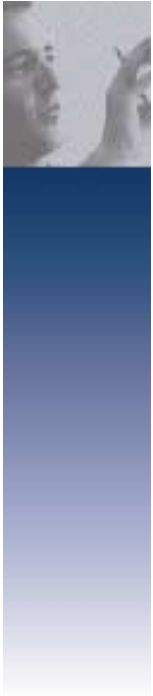
“Get Most Bang for the Buck”

- First things first
 - Do what is the most important first.

**Things that matter the most
must never be at the mercy of things
which matters least.**

Goethe





“Power to the Customer”

- Responsibility
 - Software development people
 - Customer

- Prioritisation
 - Feature
 - Architecture

The customer is always right

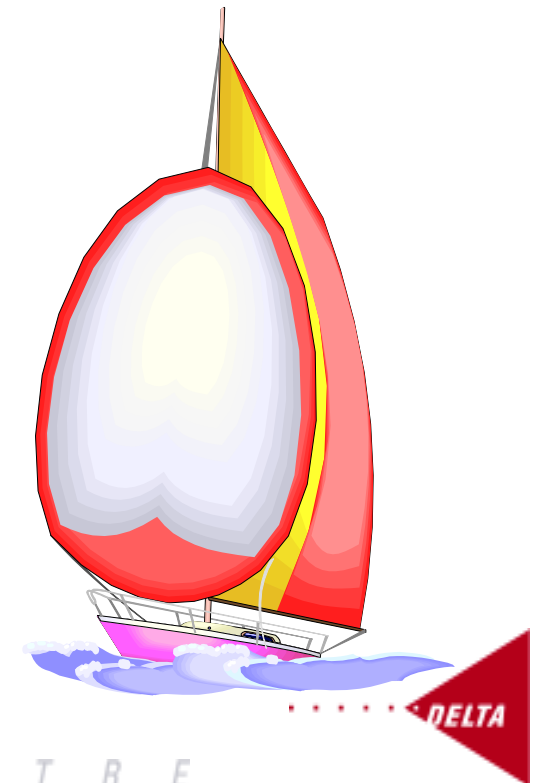


“Empowered Teams”

- Decision Speed
- Competencies must be present.



A little practice beats a lot of steering committee meetings



T H E K N O W L E D G E C E N T R E

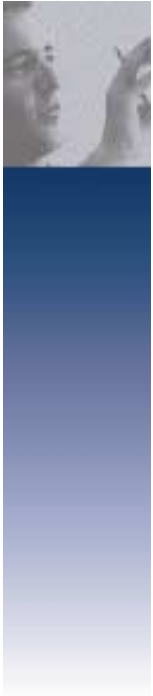


“Adaptability”

- Learning
 - Evolution / Definition
 - Ability to change
 - A specification may act as a learning obstacle

**If the map and the landscape disagree,
trust the landscape!
Winston Churchill**





“Short Life Cycles”

- Experience capability
 - When you have integrated the almost complete system 10 times, integrating the complete system is no big deal.
- Risk management
 - The faster you bring your stuff to the target, the more you will learn about the full life cycle.

**Deal with small problems
before they become big.**

T H E K N O W L E D G E C E N T R E

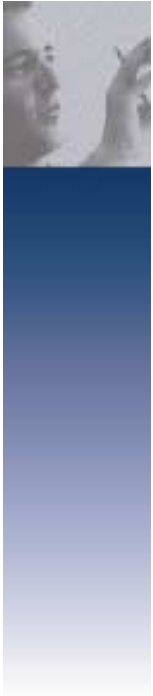




“Fast Feedback”

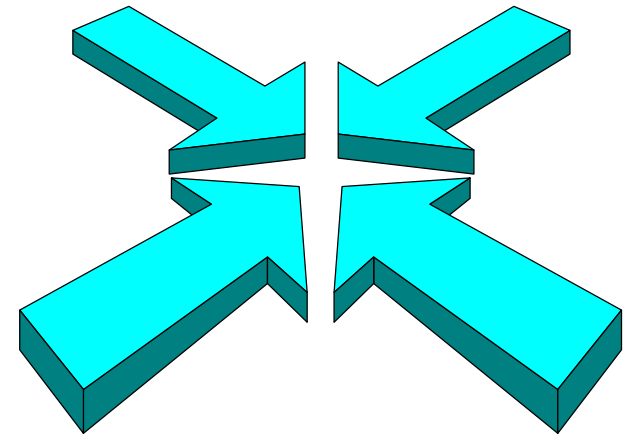
- Risk management
 - The sooner your customer see the product, the smaller mistakes are you able to make.
- At all levels
 - developer
 - feature
 - system
 - product

**It is better to light a candle,
than to curse the dark.**



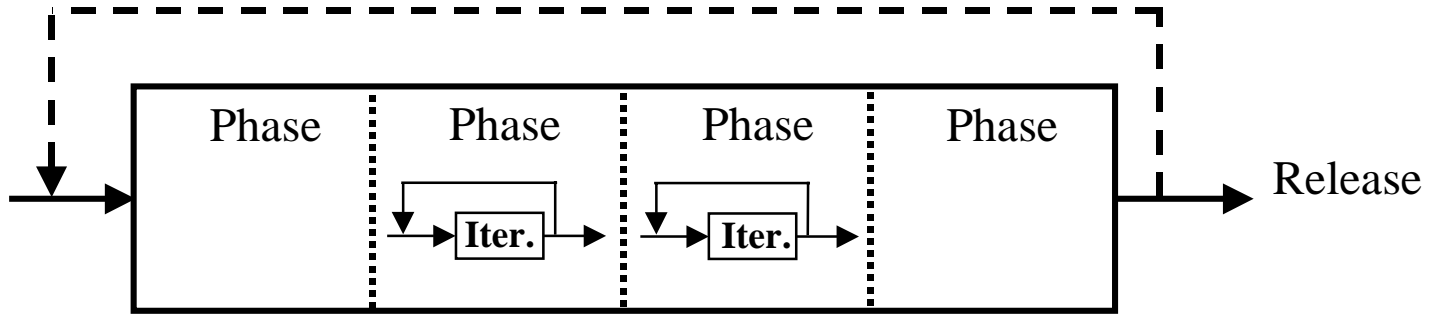
Aspects Related to Iterative Development

- Iteration
- Continuous Integration
- Configuration Management
- Requirements Management
- Project Management
- Test
- Myths and Obstacles
- Legal Perspective

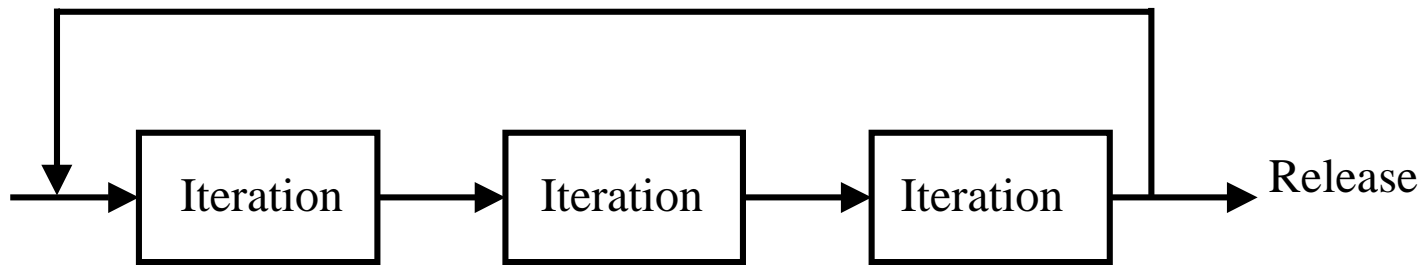




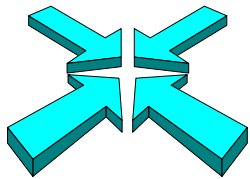
"The Iterative Life Cycle"

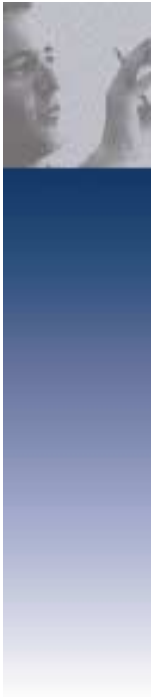


Phase-based Life Cycle Model



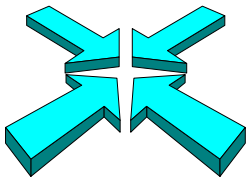
Iteration-based Life Cycle Model





“Continuous Integration”

- Personal level
 - Features
- Project level
 - Daily builds
- Product level
 - Releases



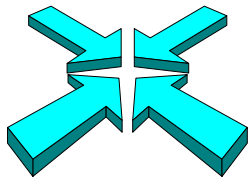
T H E K N O W L E D G E C E N T R E





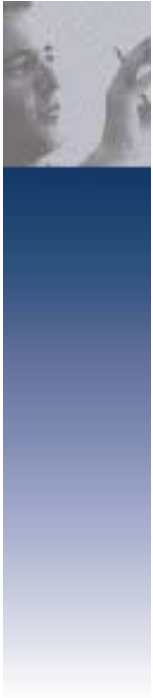
“Configuration Management”

- Even more important than in waterfall development
 - Change Management
 - Continuous Integration
 - Frequent Releases
 - Build Management



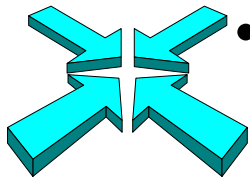
T H E K N O W L E D G E C E N T R E





“Requirements Management”

- Elicitation
 - An ongoing process
 - No need for a complete and detailed requirement specification at start
- Prioritisation
 - For the next iteration
- Measurements

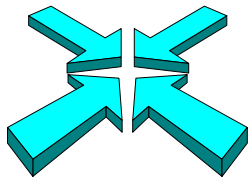


- Very different from Waterfall projects



"Project Management"

- From Waterfall to Iterative:
 - From manager to coach.
 - From "keeping the team on track" to "enabling the team to shape the track"
 - From managing detailed activities to manage in iterations
 - From "one time planner" to "continuous re-planner"



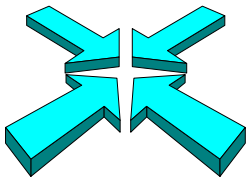
T H E K N O W L E D G E C E N T R E

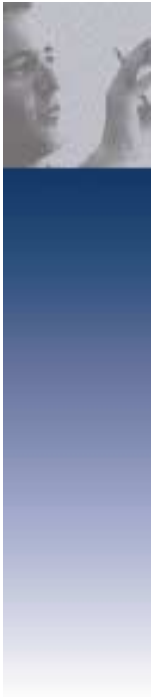


An ongoing process, not one shot at the end

"Test"

- Testing in Feature Development Iterations
 - Peer reviews
 - Feature tests
 - Test automation
- Testing in Clean-up and Reorganising Iterations
 - Test reorganisation
 - Regression test
- Testing in Release Iterations
 - Similar to waterfall project





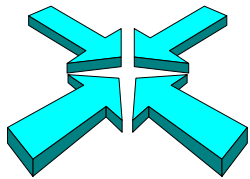
"Legal Perspective"

Contract Types

- Total Delivery (K18/K33)
- Delivery in Phases
- Time and Material Contracts
and
- Combinations

Advice to the Customer

- Define if ISD will be beneficial in the actual case
- Are you willing to accept the risk of loss of investment?
- Establish communication channels in the contract
- Operate in short periods (2-4 weeks) to minimise risk
- Formalise renegotiations in the contract
- Ownership



T H E K N O W L E D G E C E N T R E

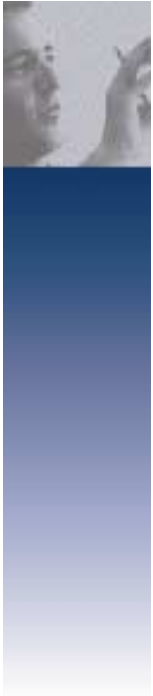




Quality and Maturity Models

- ISO 9000
- CMM
- Bootstrap
- SPICE



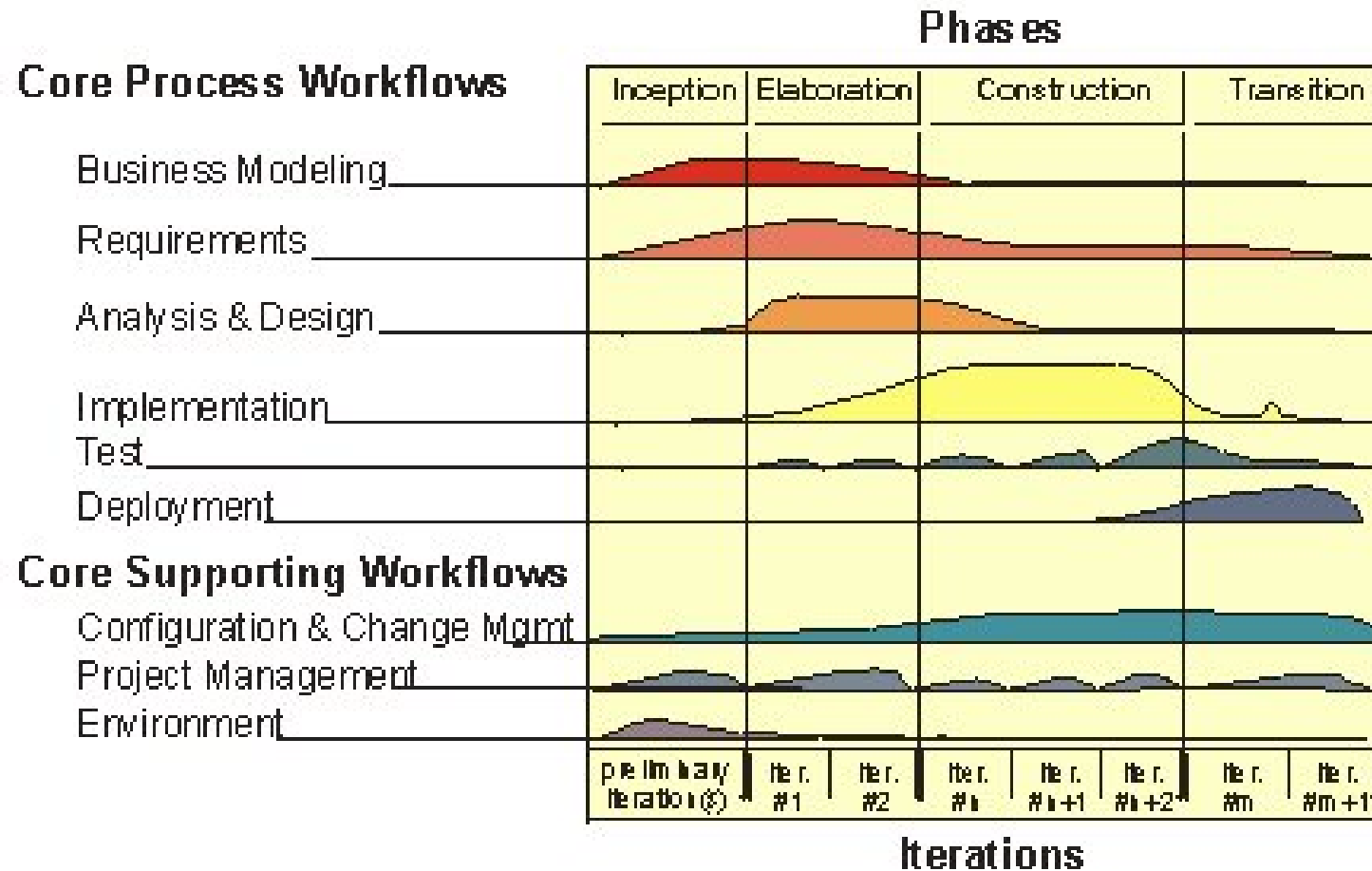


Iterative Development Methods

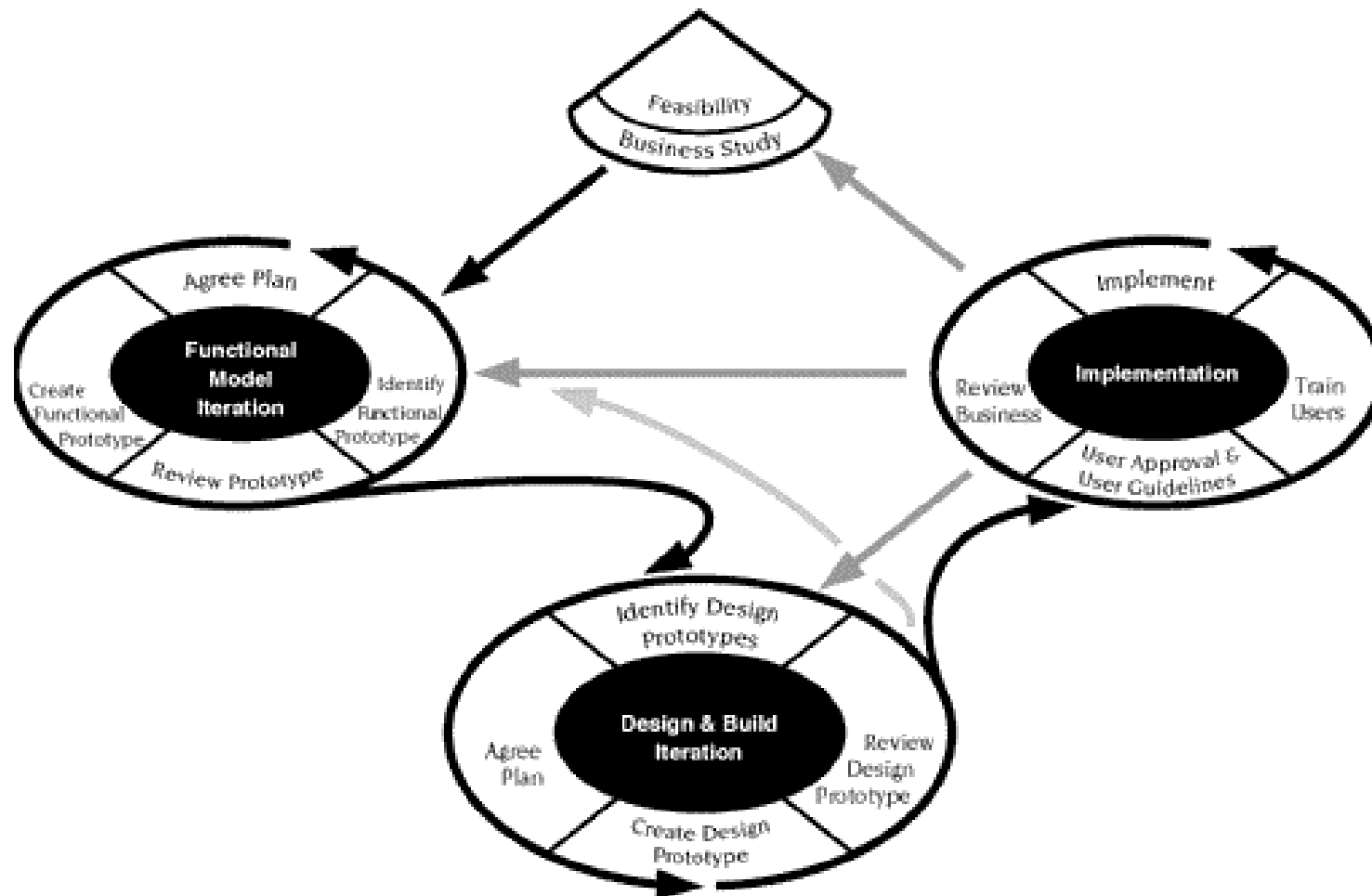
- Dynamic Systems Development Method – DSDM
 - DSDM Consortium - www.dsdm.org
- Microsoft Solutions Framework - MSF
 - www.microsoft.com/business/services/mcmsf.asp
- Rational Unified Process – RUP (Booch e.a.)
 - www.rational.com/products/rup
- eXtreme Programming – XP (Bech e.a.)
 - <http://c2.com/cgi/wiki?ExtremeProgramming>

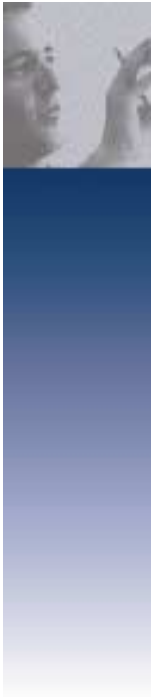


"RUP - Rational Unified Process"



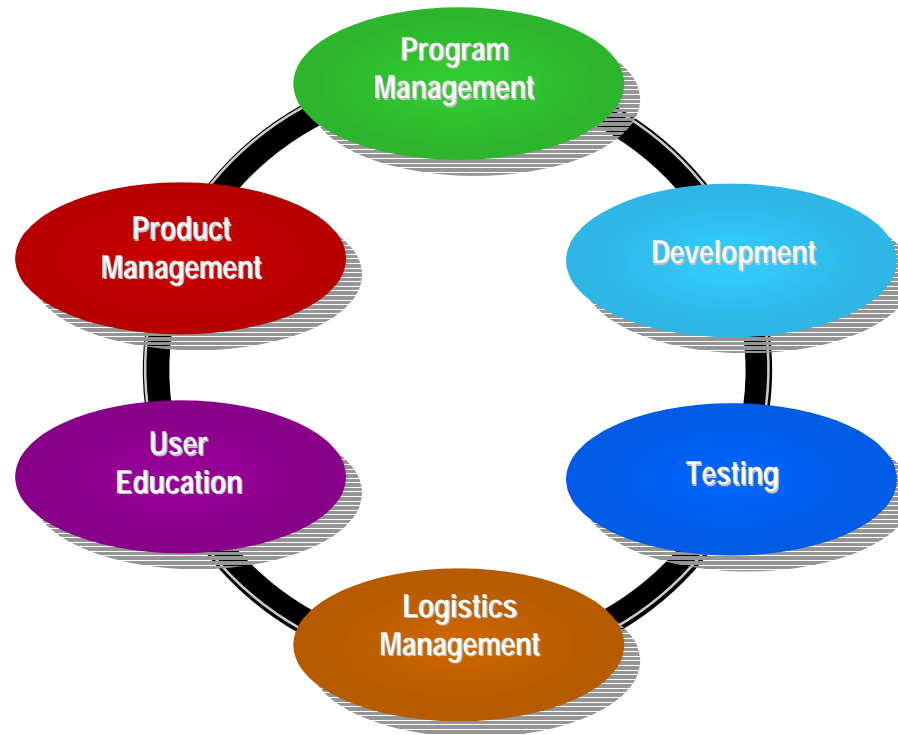
“DSDM - Dynamic Systems Development Method”

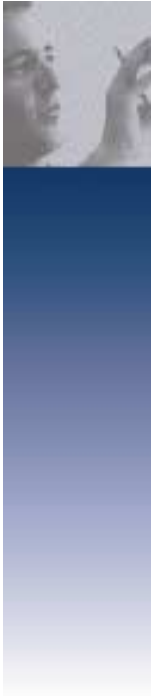




“MSF - Microsoft Solutions Framework”

- Roles and responsibilities





“XP - eXtreme Programming”

XP Values

- **Communication**
- **Simplicity**
- **Feedback**
- **Courage**

XP Principles

- **Planning game**
- **Small releases**
- **Metaphor**
- **Simple design**
- **Testing**
- **Refactoring**
- **Pair programming**
- **Collective ownership**
- **Continuous integration**
- **40-hour week**
- **On-site customer**
- **Coding standards**

T H E K N O W L E D G E C E N T R E

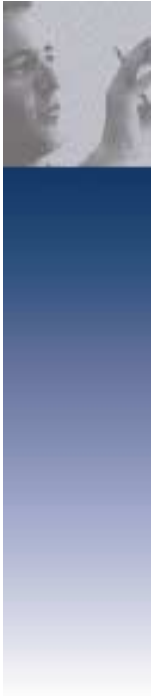




Experiences - Cases Studies

- XP @ BANKDATA
 - A MSF Case Study from Navision
 - Unibank
 - Introducing RAD at the Danish State Archives
 - Implementing an Iterative Development Process at Tellabs
 - Iterative Development at Brüel & Kjær
- and
- How this report was written





Myths and Obstacles

- Standard contracts
- Hierarchical organisations
- Mission critical software
- Large projects / Scalability
- No direct customer
- “You never know what you get”
- “ISD is hacking without a plan”





Iterative Software Development – A Practical View

www.delta.dk/iterativ:

- Forkortet udgave af rapporten
- Spørgeskema om iterativ udvikling i praksis

Pris:

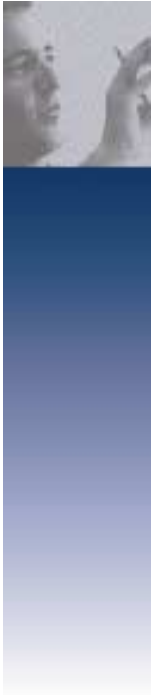
- Medlemmer af Datateknisk Forum kr 200
- Andre kr 800

Kontakt: Karin Macko, DELTA

Tel: 7219 4244, Fax: 7219 4001, email: km@delta.dk

T H E K N O W L E D G E C E N T R E





Datateknisk Forum

ERFA gruppe 25: eXtreme Programming

P.t. 29 medlemsvirksomheder

Mødefrekvens ca. hver anden måned

Næste møde 4. Juni

Medlemskab: kr 7.500 (DTF) + kr 1.500 (ERFA gruppen)

Ang. medlemskab kontakt: Karin Macko, DELTA

Tel: 7219 4244, Fax: 7219 4001, email: km@delta.dk

T H E K N O W L E D G E C E N T R E

