

July 2005



Reducing Business Risk

Through Agile Development

Fred Tingey
Head of Risk Systems
BNP Paribas



Introduction

■ Context

- What is Agile Programming ?
- Traditional vs Agile approach
- A New Way to do Things

■ What are the Risks ?

- Business Projects not IT Projects

■ Why Does Agile Reduce Risk ?

- Delivery Strategy
- Velocity
- Requirement Changes

What is Agile Programming ?

- Agile Programming is a different way to develop software
 - There are a number of methodologies sharing a common set of values that fall under the 'Agile' umbrella
 - The most popular are 'Scrum' and 'Extreme Programming'
 - Check out <http://www.agilealliance.com>
- The 'values' that make a methodology Agile are :

Individuals and interactions over *processes and tools*

Working software over *comprehensive documentation*

User collaboration over *specification negotiation*

Responding to change over *following a plan*

A New Way to do Things



The Original Version...

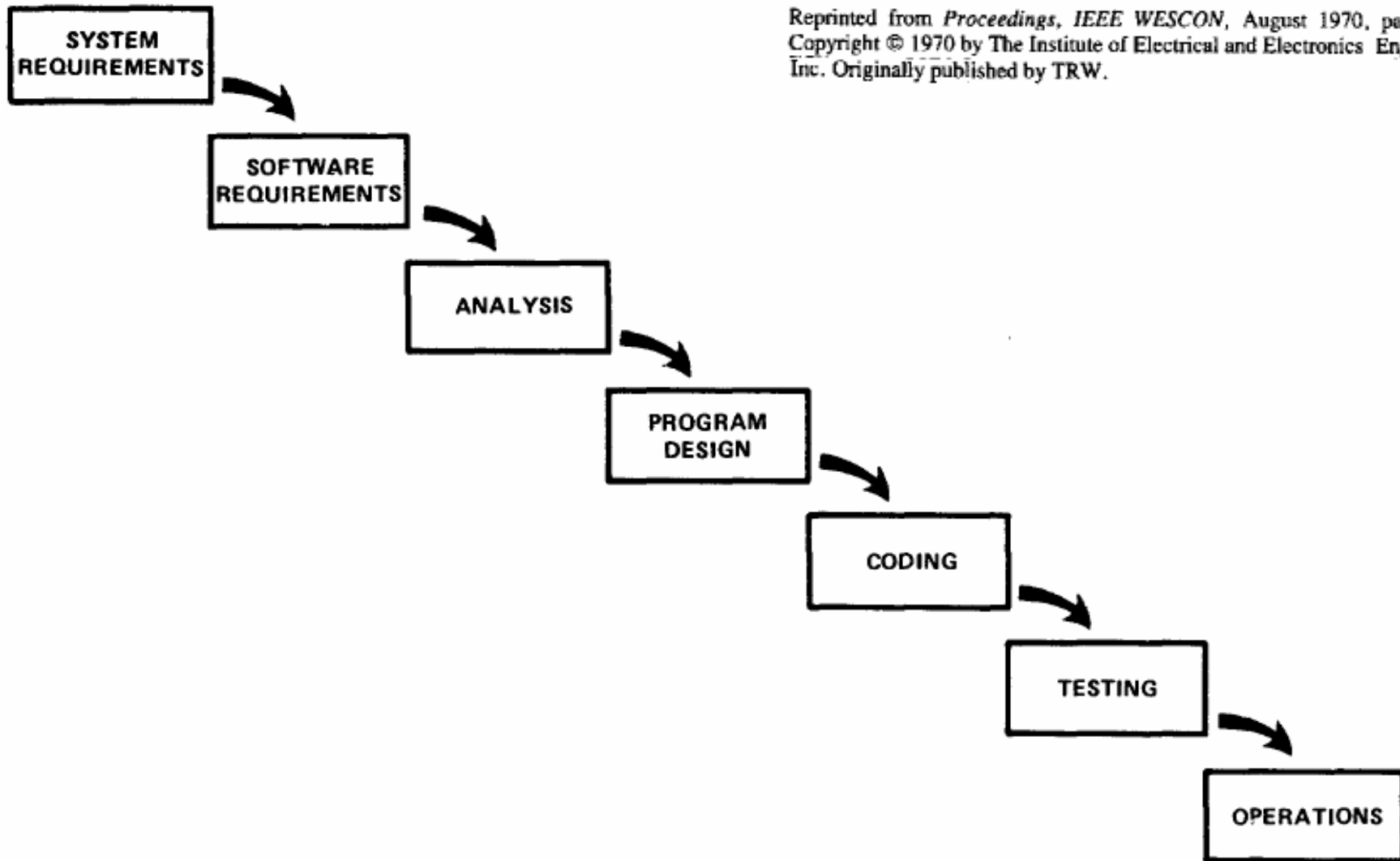


Figure 2. Implementation steps to develop a large computer program for delivery to a customer.

I believe in this concept, but the implementation described above is risky and invites failure.

And the next page goes on...

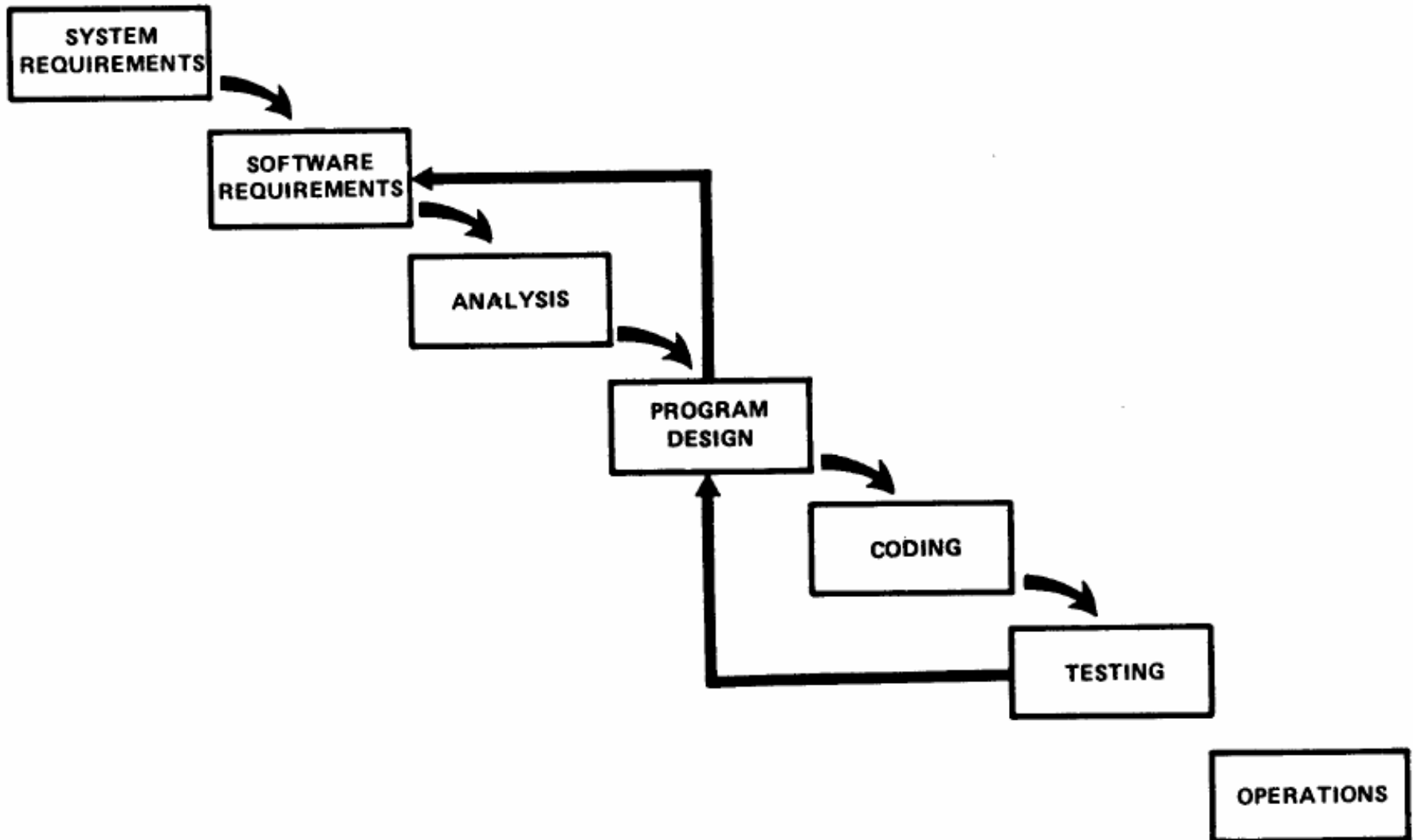


Figure 4. Unfortunately, for the process illustrated, the design iterations are never confined to the successive steps.

...To a More Realistic Model ?

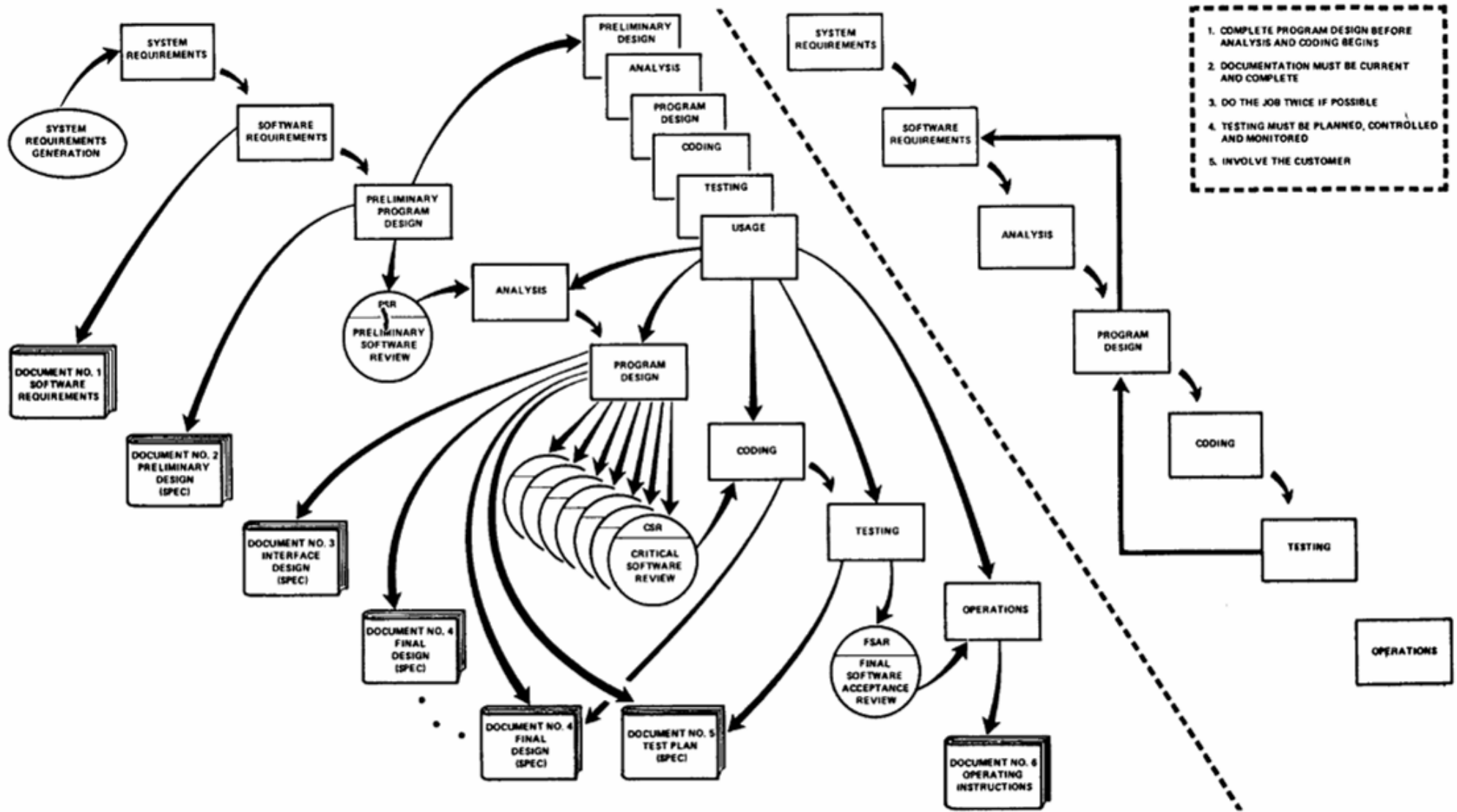
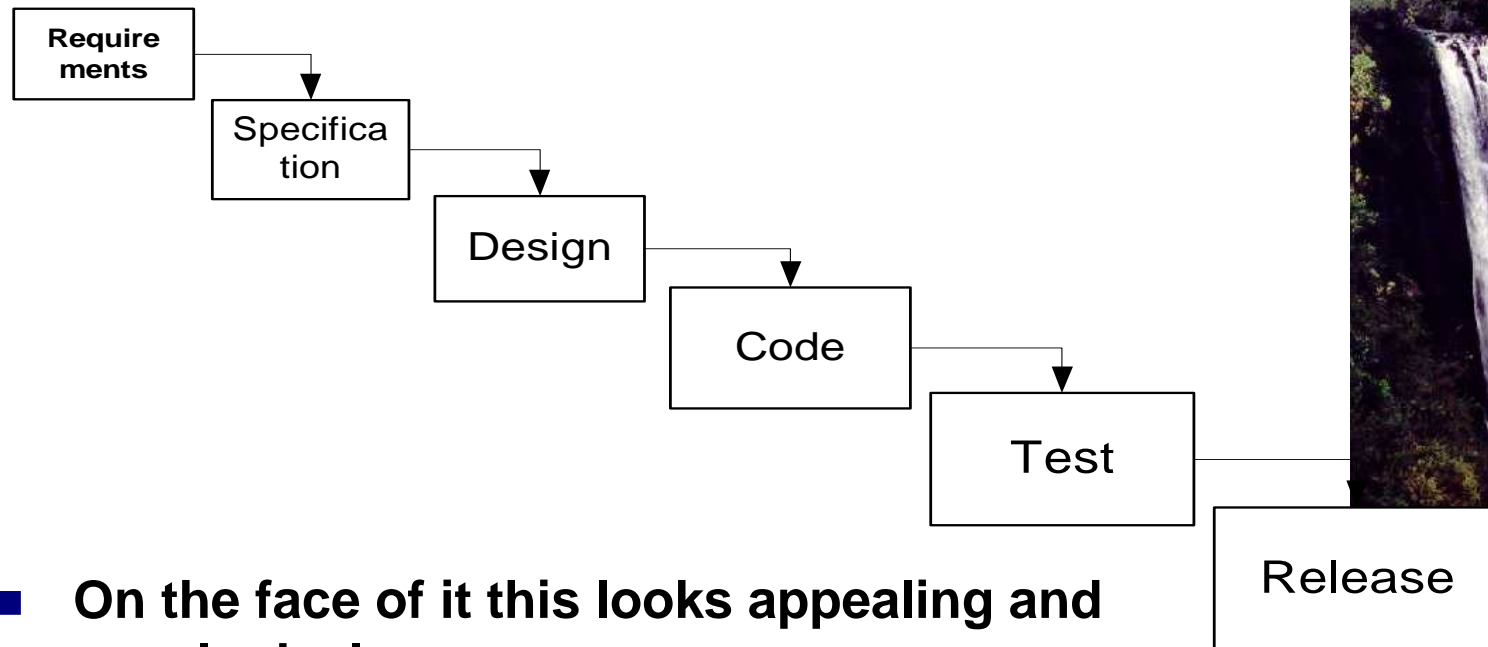


Figure 10. Summary

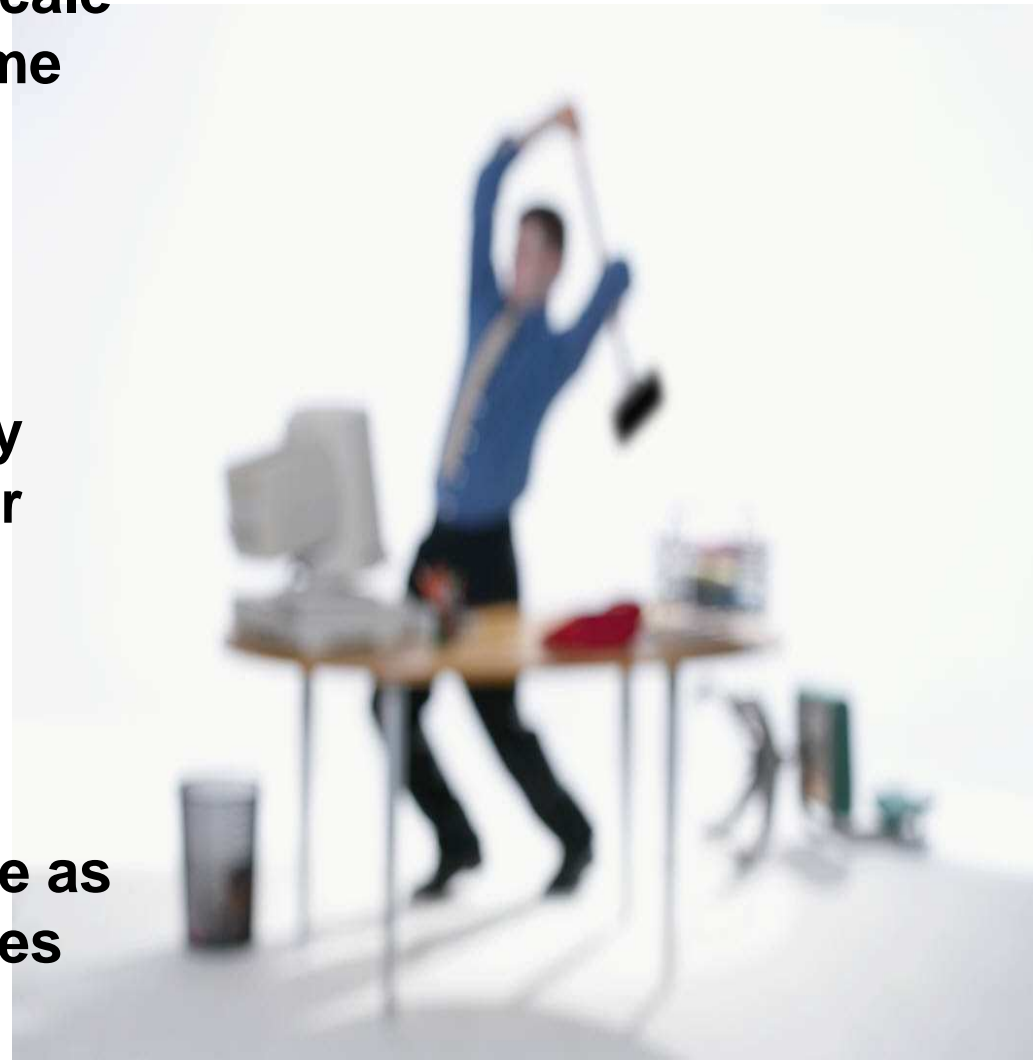
The Waterfall model epitomises the Traditional Approach



- On the face of it this looks appealing and very logical
- But there are big problems going this way e.g.
 - Very difficult and costly to change your mind or respond to change late in the life cycle
 - Do you ever get all the requirements up front and even if you do go for it how many eventually get implemented ?
 - Saving the integration and testing till the end is asking for trouble
 - You do not have anything that works or adds any value until very late in the project

Delivering value with IT to the business is hard...

- **“Some 75 percent of most large-scale IT projects fail by missing both time and budget projections ...”**
 - Mark Driver, Gartner
- **Even in successfully delivered projects: “64% of features actually delivered are either rarely or never used”**
 - Jim Johnson, Standish Group
- **Of the work executed: “Many (possibly most) organisations lose as much as 45% of their total revenues due to costs associated with low quality”**
 - Six Sigma Report



The Context

- **There is no such thing as an IT Development Project**
 - There are only business projects which need software support to a greater or lesser extent
 - Often the biggest risks on a project are related to the software development part
- **The impact of software on new and existing business processes cannot be over estimated**
 - Frequently software solutions do not take usability and workflow considerations into account until it is too late
- **The role of business analysis is to come up with solutions to business problems**
 - The last thing you want to do is to write software !

“Waterfall Does not Work”

- I will argue that traditional software development methods simply do not address the main risks involved and that is why developments so often get into difficulty

- To put it another way:

- **“Waterfall Does Not Work”**

- Fred Brooks; Author ‘The Mythical Man-Month’



In 1975, Fred Brooks looked around at the then-prevailing approach to project management and memorably declared, "Nine women cannot have a baby in a month." [Fred Brooks, *The Mythical Man-Month*, Second Edition Addison-Wesley, 1995.]

- “The primary reason project success rates have improved [doubled since 1994 to 34%] is that projects have gotten smaller. Doing projects with iterative processing as opposed to the waterfall method... is a major step forward”

- Jim Johnson, Standish Group - CHAOS report 2004

Agile Misconceptions?

- Agile means:

- “letting the programming team do whatever they want with no project management, and no architecture, allowing a solution to emerge, the programmers will do all the testing...”



Copyright © 2003 United Feature Syndicate, Inc.

Our Experience of Agile / XP

- **XP is as disciplined as any document based Waterfall approach. The disciplines work all the time.**
 - **An Iterative approach forces real clarity from the business on what they want, why they want it and where the priorities really lie**
 - **Regular release cycles (Iterations) introduces a rhythm to the development process; it is quick and easy to spot when things are not working right**
- **A Training / Mentoring program is required to switch paradigm from Traditional to Agile.**
 - **The business have to buy in to the approach and be taught to be Agile as well**
- **Increased quality; Software has less bugs and is more 'fit for purpose'.**
- **The software delivery cycle has to fit into the overall business project delivery cycle.**

Continuous Integration...

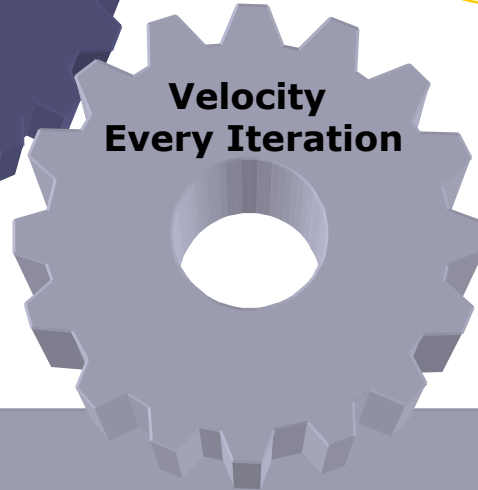
Team-Level Planning



Daily Standup Meeting:
15 minutes

Each team's member answers 3 questions:

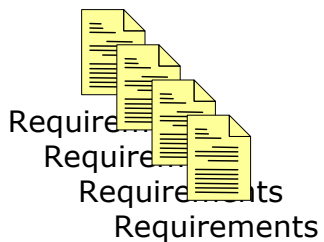
- 1) What did I do since last meeting?
- 2) What obstacles are in my way?
- 3) What will I do before next meeting?



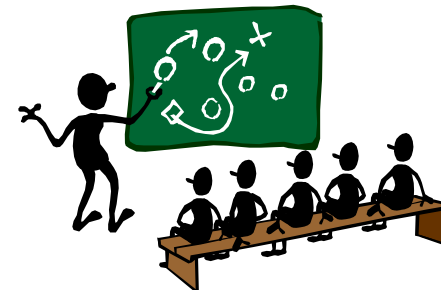
Prioritised Iteration Scope



Working Software Delivered

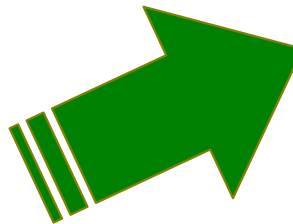
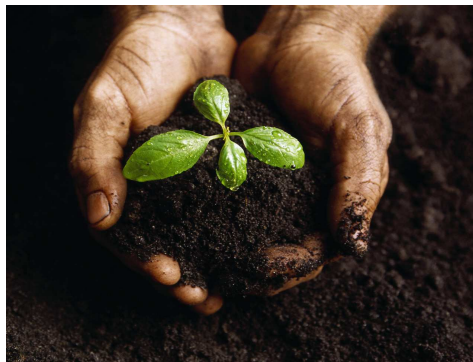
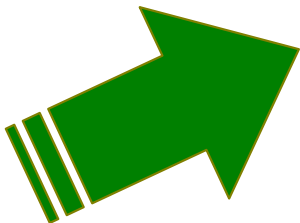


Prioritised Requirements & Features "Backlog"



Delivery Strategy: Start small and grow

- Focus on addressing and lowering delivery risk
- Divide application into bite-sized chunks of functionality
- Integrate continuously; prove delivery organisation, tools and methods
- Begin testing early
- Work iteratively and deliver regularly
- Where possible build out from proven frameworks



What are the Risks ?

■ Estimation Risk

- It is very difficult to correctly estimate the duration or complexity of development or design tasks.

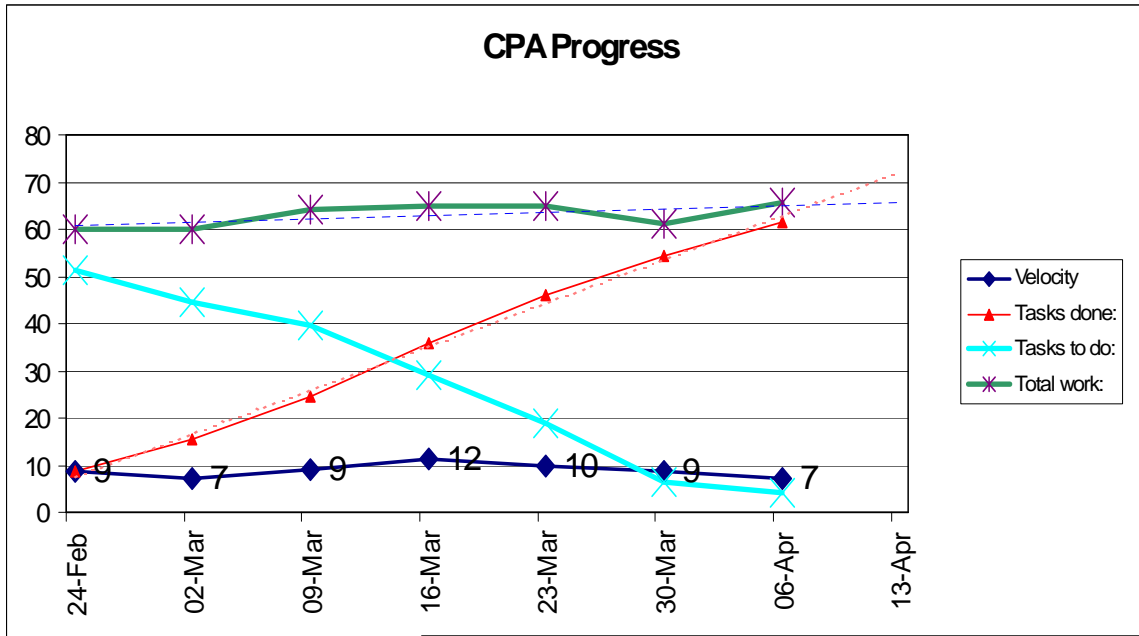
■ Changing Requirements Risk

- Development projects must be able to adapt to changes (in requirements or priorities) throughout their lifecycle.
- End users often do not see the software until it is 'finished' and too late to include necessary feedback
- The world moves on and business needs evolve

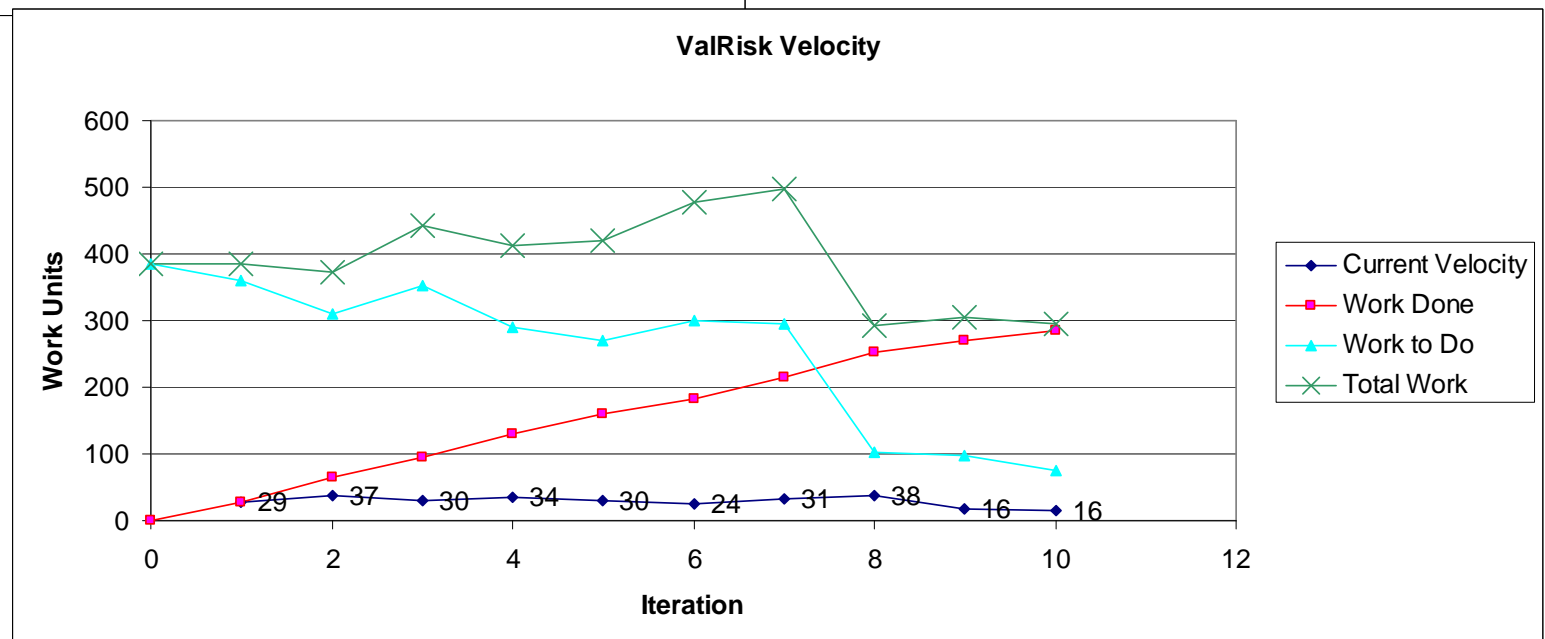
■ Integration Risk

- Individual parts of a system may work but it is virtually certain that when you put them all together the system will not.
- *And that includes the end users as part of the business process.*

Iterative Velocity Measures



- A Couple of Examples of Velocity on real projects
- ‘Work Done’ relates to Stories that have been delivered and tested
- Velocity measures real progress



Why does Agile reduce Risk ?

■ Estimation Risk

- Through the measure of Velocity per Iteration it does not take long to gather enough empirical evidence (based on fully tested working software) to make estimates much more accurate.**
- You know how many things you can do each Iteration and you know how much is currently left to do.**

■ Changing Requirements Risk

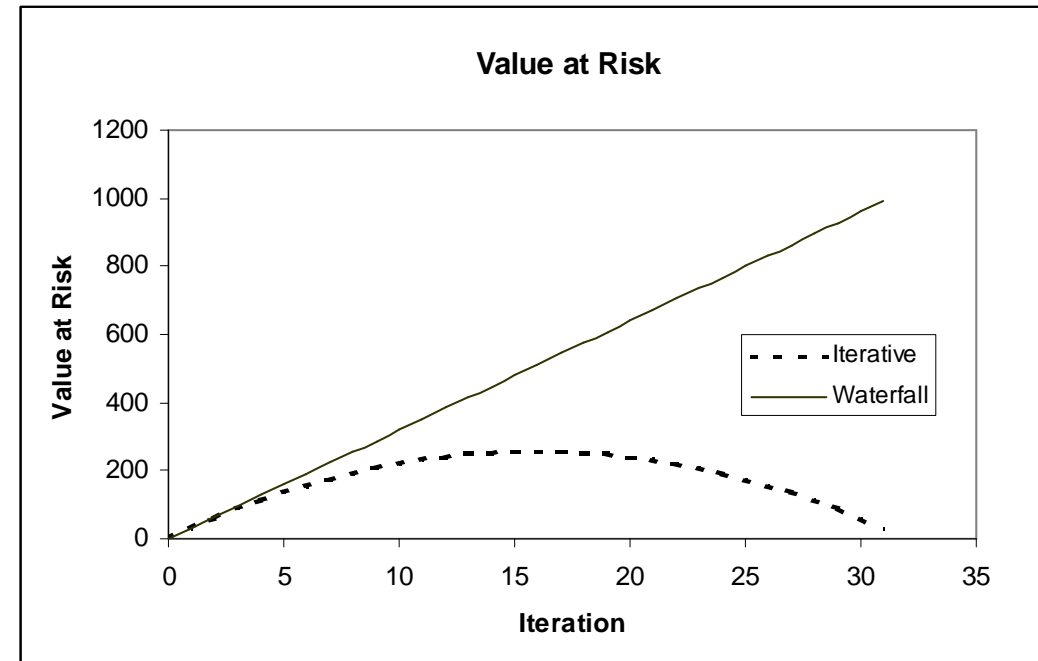
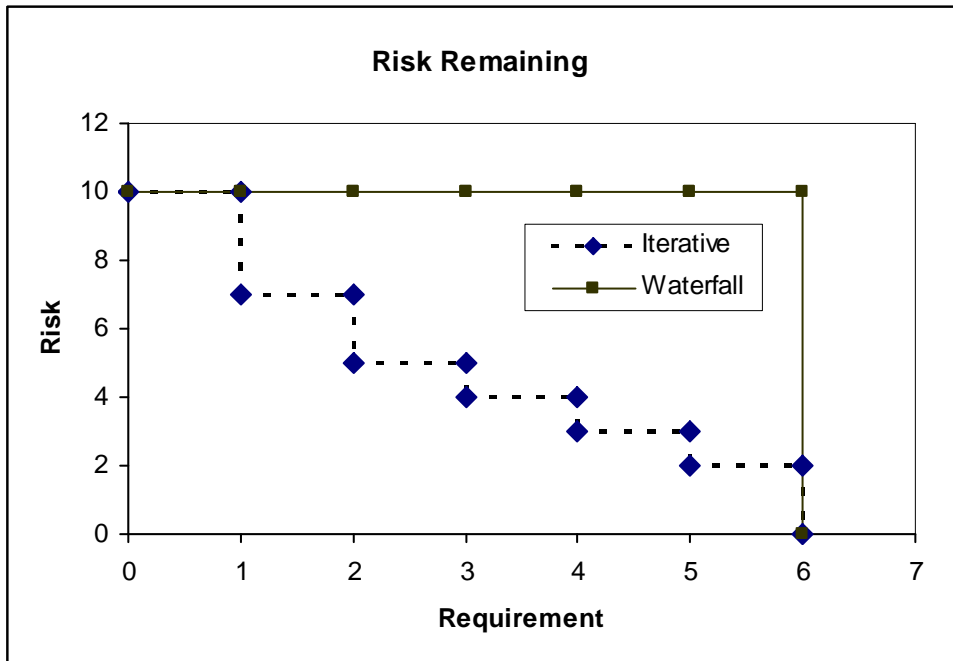
- Breaking the project down into Iterations means that changes to the plan can be accepted very easily.**
- At the end of each Iteration you have a working piece of software that can be tested by the users. The more the users see the more ideas they have to improve it and add business value.**

■ Integration Risk

- This is virtually eliminated through the practices of Continuous Integration and Test Driven development.**
- Working software means that business processes can be fully included in the integration and testing as you go along.**

Iterative Methods reduce Requirements Risk

- Feedback on requirements
 - Reduces the Uncertainty remaining...



- Value at Risk
 - Waterfall increases until the end
 - Iterative reaches a lower maximum

How does XP fit into a Large Enterprise ?

■ Start small

- The change to XP is non-trivial
- Train up a small team and then grow into other areas
- Start with 'User Stories' and 'Velocity'

■ Fitting Business Analysts in

- BAs represent the end users
- They must be part of the project team
- They are the principal source of detailed acceptance tests

■ Enhancing XP for the 'Enterprise Culture'

- Dealing with existing production systems
- Staging releases and the UAT phase
- Breaking specifications into independent Stories / Features
- Business Process Re-engineering must be addressed
- The 'feasibility stage'
- Roles and responsibilities

The Challenges Remaining

- **To convince the sceptics**
 - Gather demonstrable proof that Agile is better
 - Implement some of the Best Practices on individual projects rather than all of them at once

- **For us the concept of the 'Agile Enterprise'**
 - XP Focuses on the development for a project; this is good but not quite sufficient in a large complex organisation
 - Addition of rules for business process re-engineering; Documentation for communication; Roll Out planning etc.

- **Agile Contracts**
 - Moving from Fixed Price to flexible Incremental Delivery

- **Agile Certification**
 - International standards to compete with CMMi

Conclusion

- **Agile methods are now established beyond doubt**

- **Agile Means**
 - **The Core Values**
 - **Iterative Development of Business Functions**
 - **Repeatable Testing and Automation**

- **Agile Approaches reduce Business Risk**
 - **Theoretically this makes sense**
 - **Practically we have seen this at BNP Paribas**

- **Remember :**
 - **“Waterfall Does not Work”**
 - **“Velocity Measures Real Progress”**



Questions ?
