Bootstrapping Agile – Practice by Practice

Daniel Poon



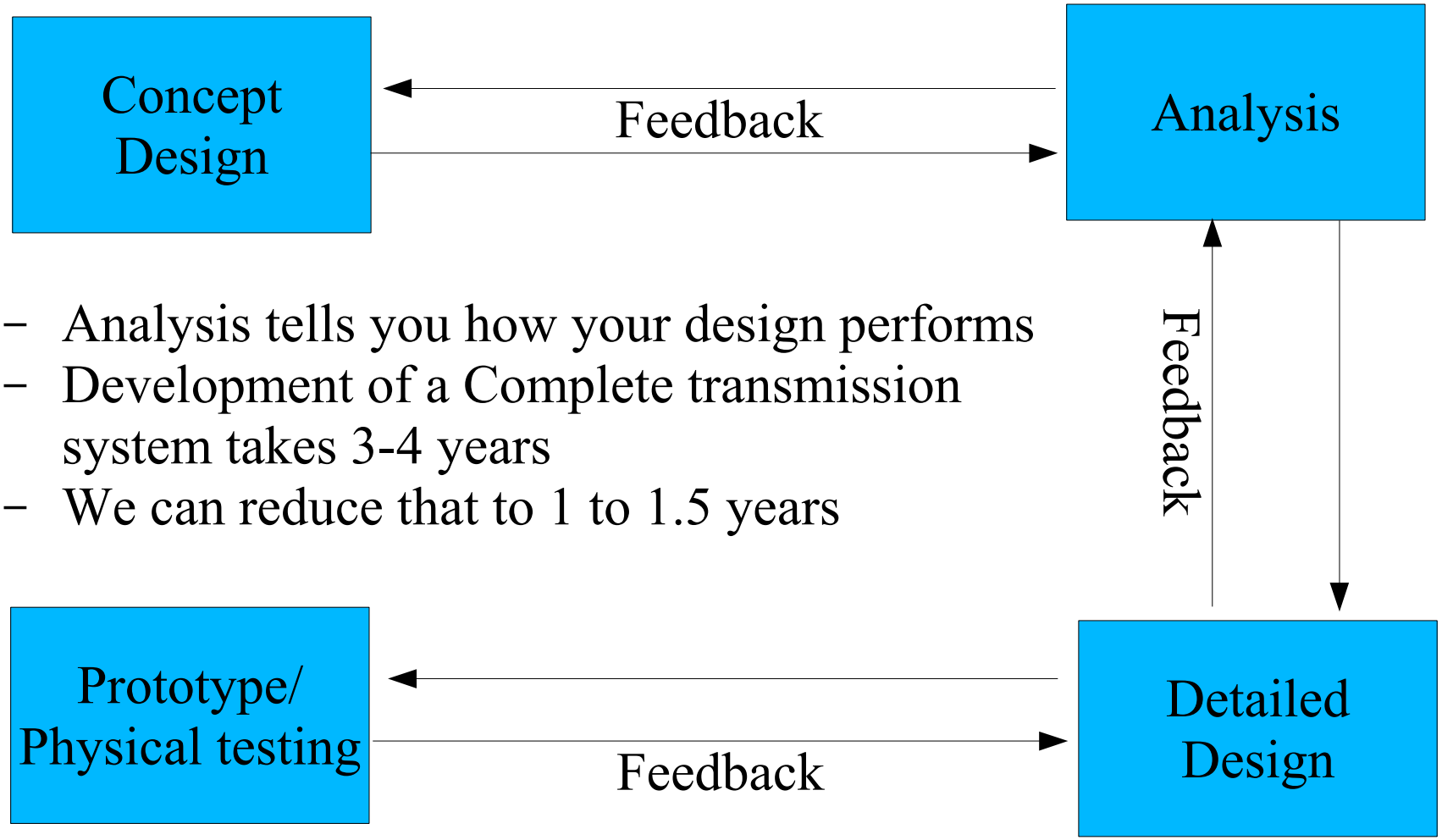
How and why did we go Agile?

- Take a look at our business and the challenges it faces
- Take a look at our solution
- Lessons learned

Romax Technology Ltd. & *RomaxDesigner*

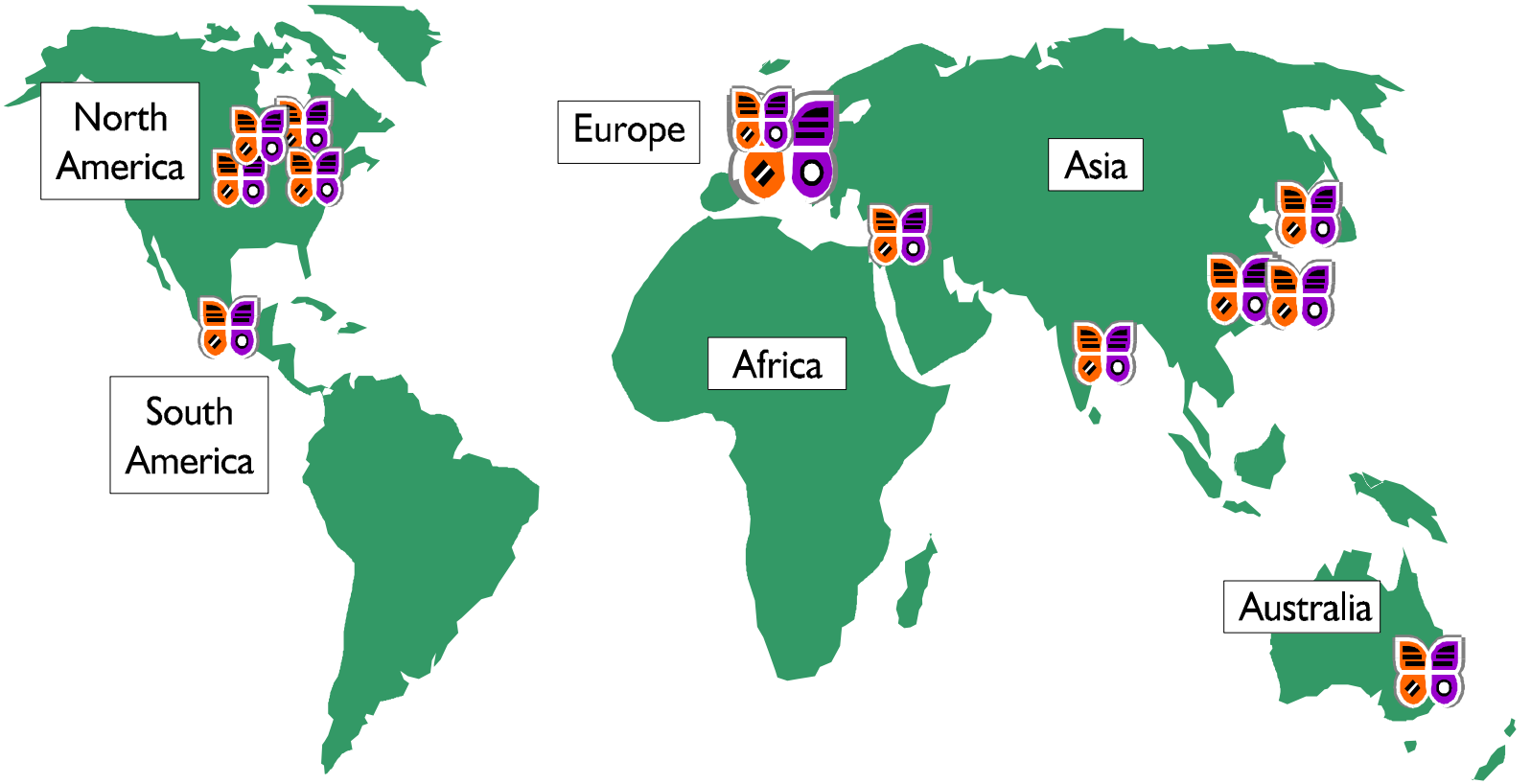
- Our value proposition
  - *RomaxDesigner* enables us to predict the system wide behavior of transmission systems beyond what was previously possible

VPD (Virtual Product Development)



- Analysis tells you how your design performs
- Development of a Complete transmission system takes 3-4 years
- We can reduce that to 1 to 1.5 years

The Challenge of a Global Market



- 95% of sales from export
- 75% of sales from Asia and USA

Technical Challenges

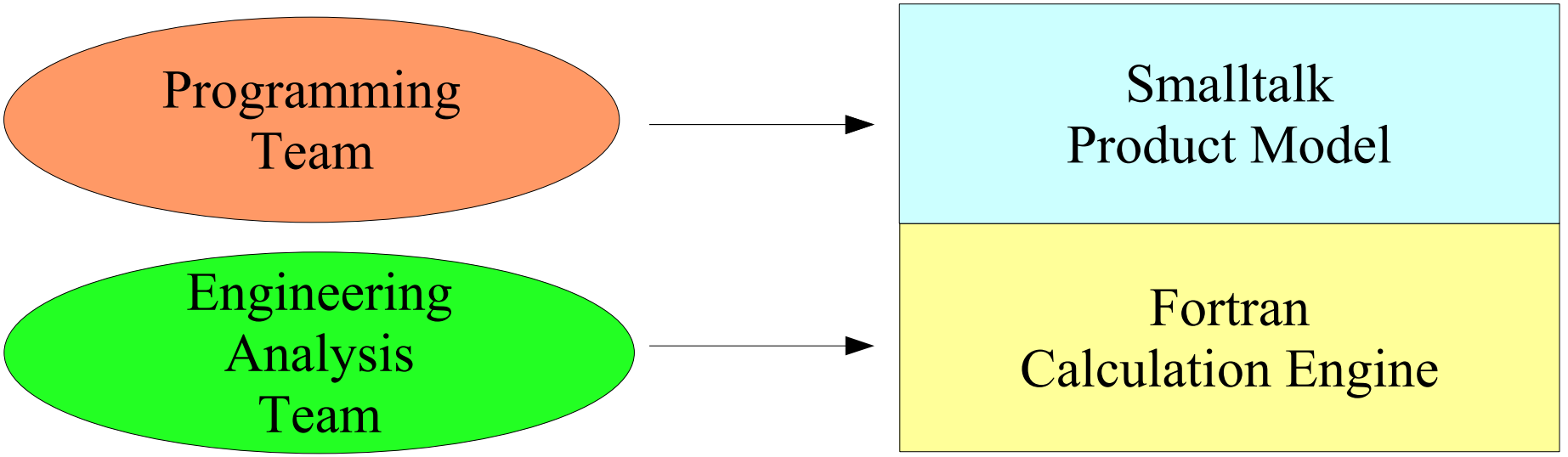
- Customers want ever more complex simulation
- Greater complexity puts greater demands on performance
- The market demands ever shorter lead times and shorter development cycles
- How does Agile help?

Legacy Code



Fragmented code base

Legacy Architecture



- Difficult to optimise and difficult to modify
- An Integration Bottleneck, making it difficult to release frequently and respond to the market

Legacy Process

Design Engineers

Automotive projects

Too busy to do testing

Engineering Analysts

Many projects

Consultancy/ Programming

Programmers

Lack of domain knowledge

Conventional MIS background of little use

Teams split into functional departments

Legacy Attitudes

Design Engineers

"This is the way we do design"

Engineering Analysts

"Matlab does everything I want"

"Fortran is best for analysis"

Programmers

"We ought to design in UML "

"I want to learn Java/C#"

Each is interested in their own bit – no one wants to look at the whole picture

Unsustainable Cash Flow

- 1 month cycle time
  - 12 people * 1 month
  - 1 man year of WIP
  - £0.1M in costs
  - **Locked away for 1 month**
  - Potential sales?
- 18 month cycle time
  - 12 people * 18 months
  - 18 man years of WIP
  - £1.8M in costs
  - **Locked away for 1 ½ years**
  - Potential sales?

The cycle time = time for your investment to become liquid

Does this sound like an organization asking for change?

- Many obstacles
  - Including hostile attitudes
- No one willing to act
  - But may be a few people are willing to be guinea pigs

Initially tried telling managers what we could do if we went Agile

- Read about XP
- Presented ideas to Management
  - "So what"

Read C2 Wiki / *XP Explained*

Present XP to the management team

Read *Developing Products in Half the Time*

2001 2002 2003 2004 2005 2006

It was better to tell them what I wanted

- Ask for political support
  - To protect your back
- But don't ask or expect material help!
  - They won't want to pay for outside help
  - Why should they, its what they pays you for!

Talk to Managing Director offline

2001 2002 2003 2004 2005 2006

Start Small: Pair-Programming with Domain Experts is a good ice breaker

- Only needs buy-in from two people
- Microcosm of the agile process
- Programmer + Domain Expert = Self Sufficient
  - Cannot be sabotaged from outside



Programmers take to TDD easily

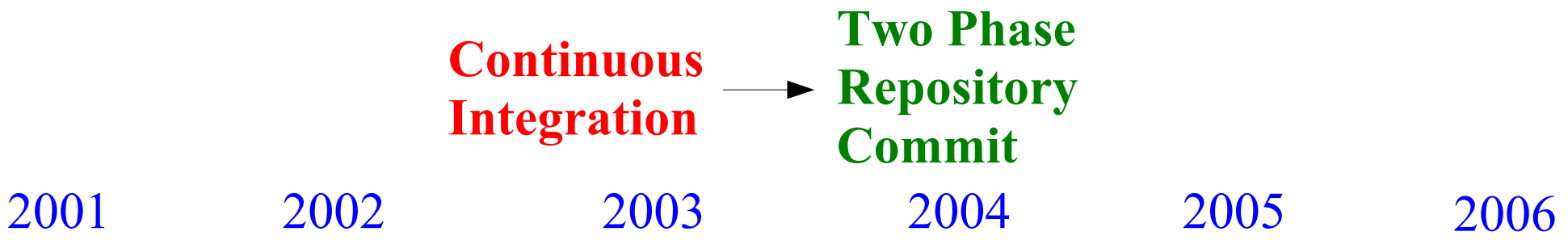
- When working with a legacy code base, write tests to compare previous results
- Gives stability and confidence
- Taught us how to test
- Now writing finer grain tests



2001 2002 2003 2004 2005 2006

Can Continuous Integration be done with Legacy Code?

- Lack of Unit Test coverage
  - Attempt at Continuous Integration failed
- Automatic, two phase, repository commit
  - Code branch acts as a staging area
  - Acceptance Tests are automatically run
  - Code is committed automatically if they pass



Use technical practices to drive management practices

- Tests = Metrics
- Fast build = Feedback
- So you have a reason to have meetings

Burndown Charts

Test Status Board

Stakeholder Steering

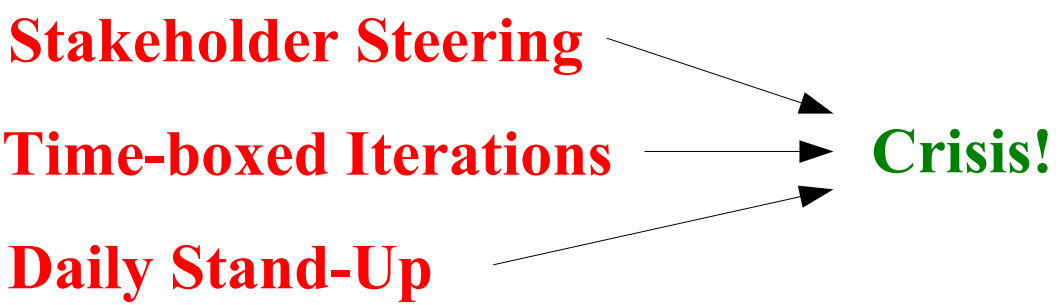
Time-boxed Iterations

Daily Stand-Up

2001 2002 2003 2004 2005 2006

Wait for a crisis!

- Customer is promised non-existent feature in one months time
- So do it as *"Real Iteration 1"*
- Used crisis and our subsequent response as a catalyst for full Stakeholder buy in



2001 2002 2003 2004 2005 2006

SCRUM's management techniques

- Scrum's 1 month cycle works well
  - Burndown chart is a substitute for shorter iterations
- Scrum's granularity works well for where we are at the moment
- Scrum's continuous re-estimating works well

Choose a System Metaphor that Managers & Domain Experts can relate to

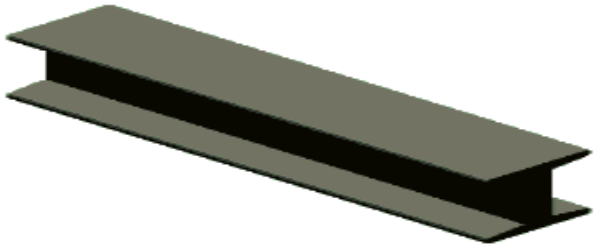
- "*Make it Run, Make it Right, Make it Fast*" mean little to Managers & Domain Experts
- You need to describes the problem in a way that all can understand – the so called *System Metaphor*

System Metaphor

2001 2002 2003 2004 2005 2006

The *Body-on-Frame* System Metaphor

- Two seemingly orthogonal requirements
  - Torsional Rigidity
  - Passenger space
- Two solutions
  - The Frame provides torsional rigidity
  - The Body provides passenger space
- Strength comes from material selection



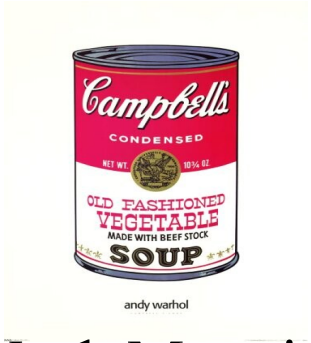
Strong Material



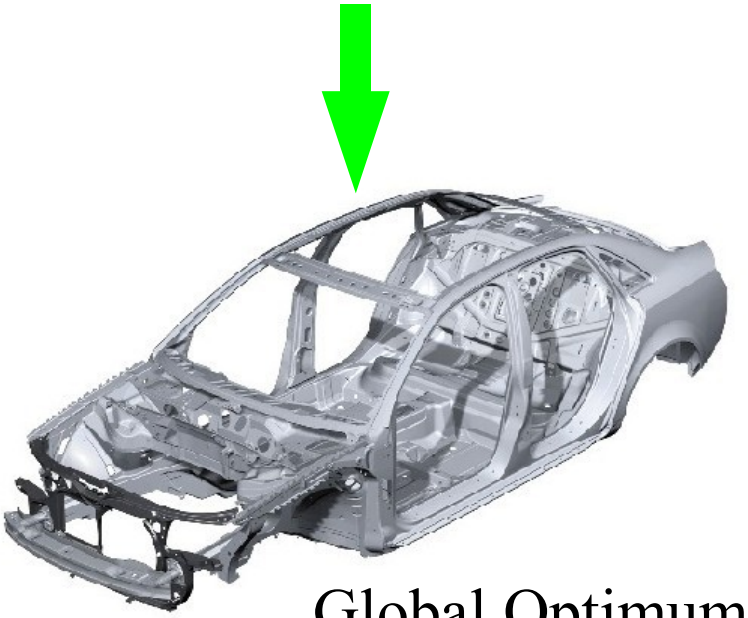
Local Optima

The aim is a *Monocoque System*

- But requirements are not orthogonal
  - Torsional Rigidity
  - Passenger space
- One solution
  - Single structure integrates torsional rigidity with passenger space
- Strength comes from disposition of material

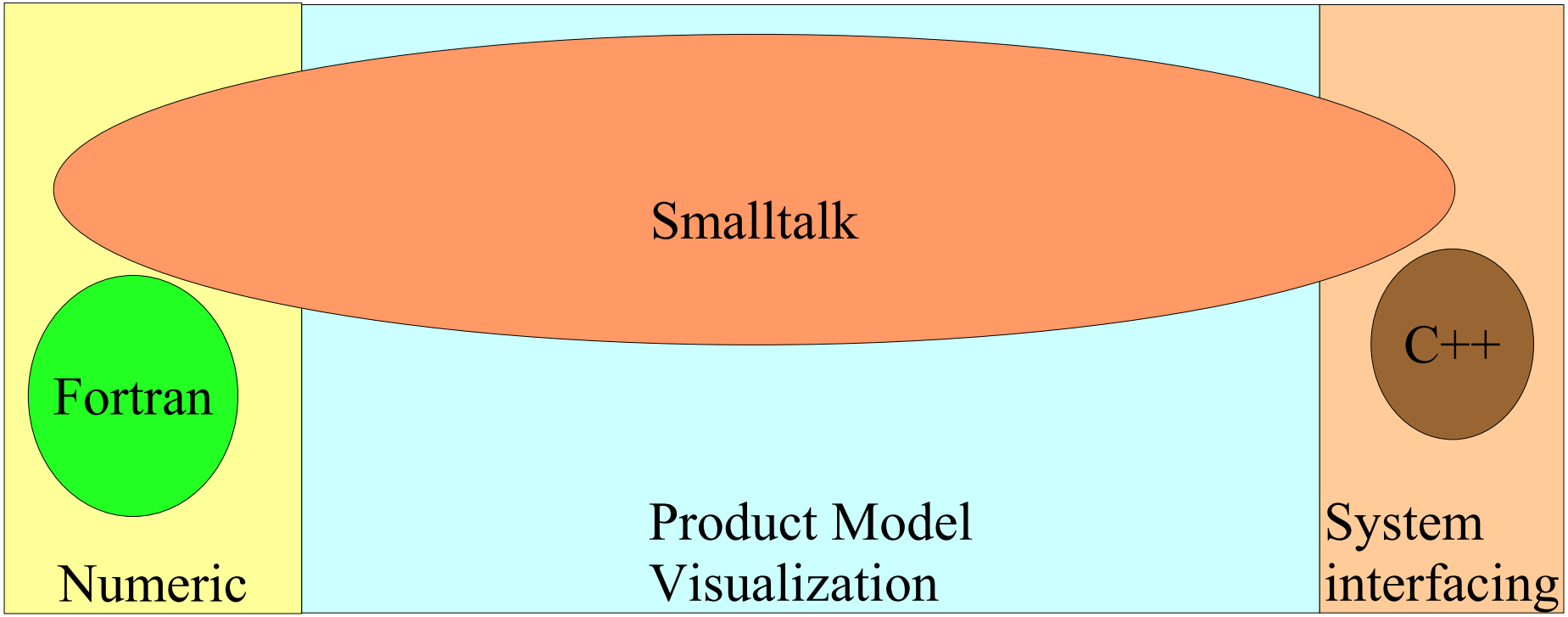


Weak Material



Global Optimum

Commercial Drivers for Shift Towards Monocoque Architecture



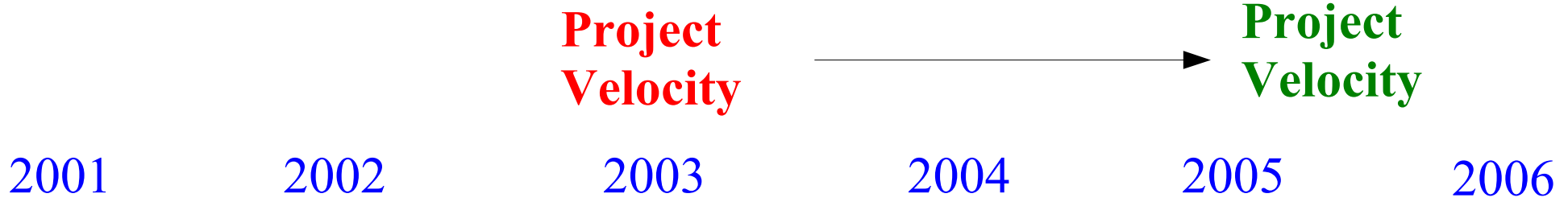
- Easy to modify and globally optimise
- May be continuously integrated

DIY approach to Agile

- Pros
  - No Capital
  - When practice finally clicks, the team will fight to keep it
  - Sense of personal and team achievement
  - No worry that the consultant will desert you
- Cons
  - Lots of Time
  - Without outside guidance, lots of painful failure
  - Frustration when things go wrong
  - You need to stay with the same team for a long time

Almost every practice we introduced failed first time around!!

- Be patient, time the introduction of new practices carefully
  - First try the simplest thing that could possibly work
    - When it breaks, fix it
  - Start with a rigid format
    - Then evolve to a problem solving format



Separate Team Lead/Scrum Master

- Split management tasks between them
- Gives you the space to make mistakes
- Learn a few coaching skills
  - very cheap if done outside work

Team Lead/ Scrum Master Roles Formally Separated

2001 2002 2003 2004 2005 2006

Large Staff Turnover

- Is it a problem, or an opportunity?
- Do people leave because you are implementing Agile, or because of the crisis you are trying to solve?
- Is a crisis the only time you can implement Agile?

Office Move

Large Staff Turnover

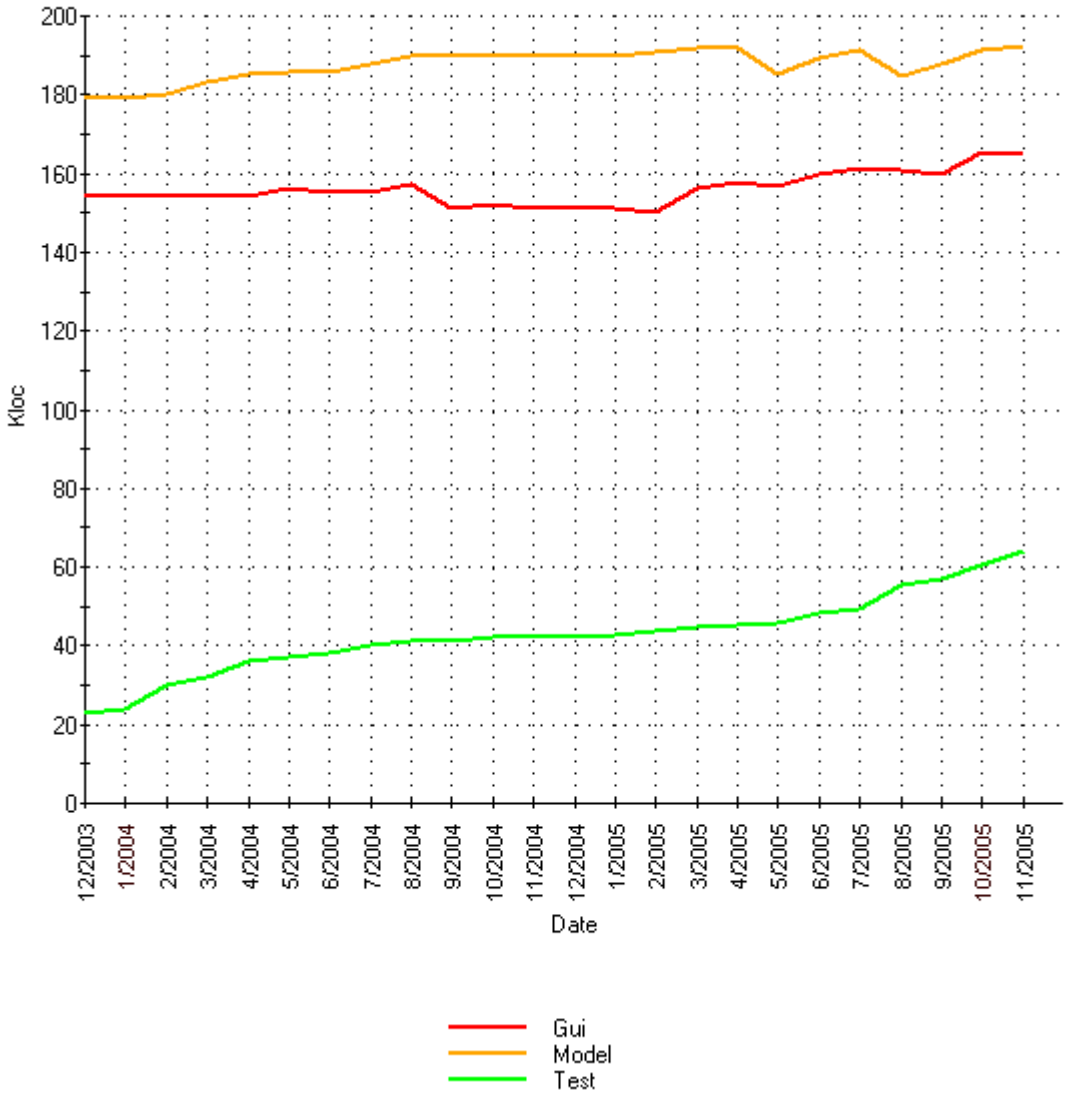
2001 2002 2003 2004 2005 2006

Retrospectives

- Failed several times to run a productive Retrospective
- Only went well when we go outside help



(Incomplete) Code metrics



- No test code before 2001
- Considerable functionality added since 2001
- Amount of GUI & model code has not increased significantly
- Test code is increasing

Still so much to do...

- Can we automatically test the GUI?
- Can we get rid of *stabilization iterations?*
- Can we speed up the build?
- What happens if our team grows anymore?
- How can we split the team effectively?
- Why don't Stakeholders collaborate together?
- Do Stakeholders realize in every planning meeting they are deciding how to spend over £50K?

Any Questions?



Thanks to
Gareth Owen, Andrew Smith, Sean Akers, Mark Eccles, Chris Halse, Richard Lord, Chris Bailey, Andy Poon, Jamie Pears
and everyone else at Romax,
plus Giovanni Asproni and Rachel Davies