Dr. Keith Collyer

Session

Innovate2010

The Rational Software Conference

Let's build a smarter planet.

Our world is becoming more complex everyday...

162 million

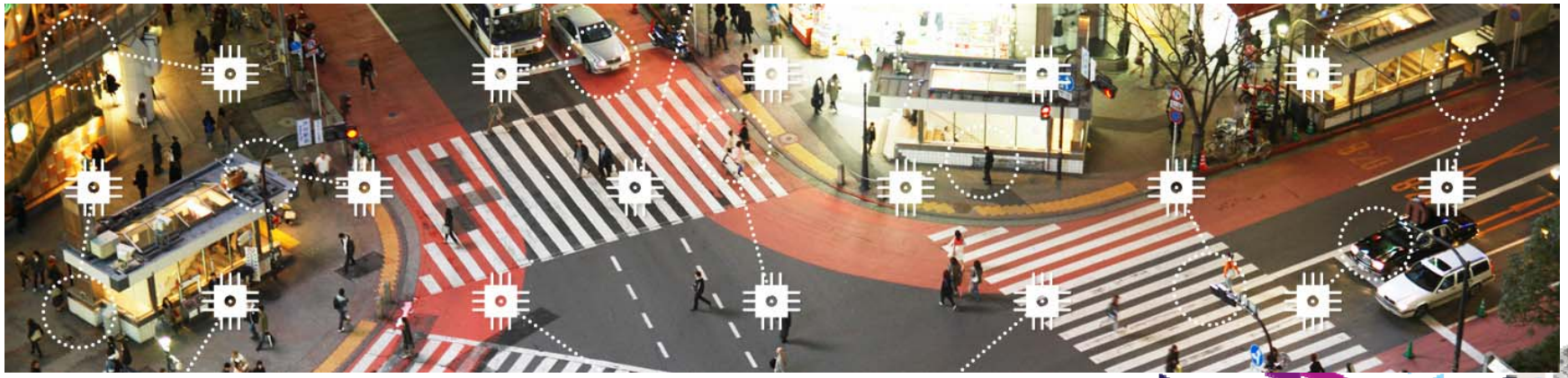
Almost 162 million smart phones were sold in 2008, surpassing laptop sales for the first time.

90%

Nearly 90% of innovation in automobiles is related to software and electronics systems.

1 trillion

Soon, there will be 1 trillion connected devices in the world, constituting an “internet of things.”



...and the challenge of managing this complexity has never been greater

KAC1

48 months

Five years ago, a typical manufacturer's concept-through production cycle time was 48 months. Within four years, that time dropped to 18 months—with a goal of reaching a 12 month cycle within the next year.

$300 billion

66% of software products are deemed unsuccessful, costing the industry nearly $300 billion annually.

42,000 units recalled

More than 42,000 defibrillators had to be recalled in 2007 due to faulty software.

Slide 3

KAC1 Replace with Toyota example

Keith Collyer, 3/19/2010

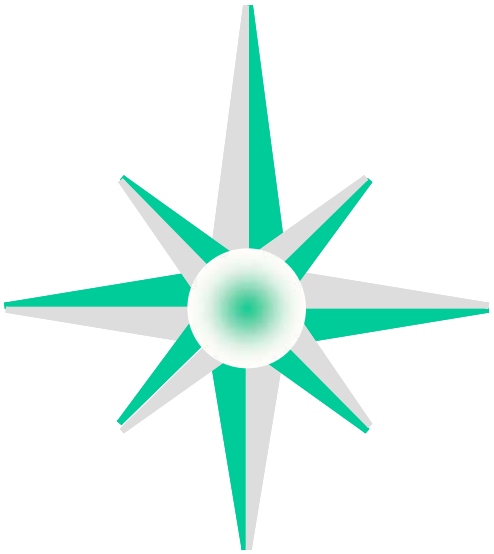
4 Principles for Effective Requirements Lifecycle Management

Recognize the needs of all stakeholders

N

Promote the use of a consistent process

W



E

Use structure to manage complexity

S

Encourage collaboration across the lifecycle

Principle 1: Recognize the needs of all stakeholders

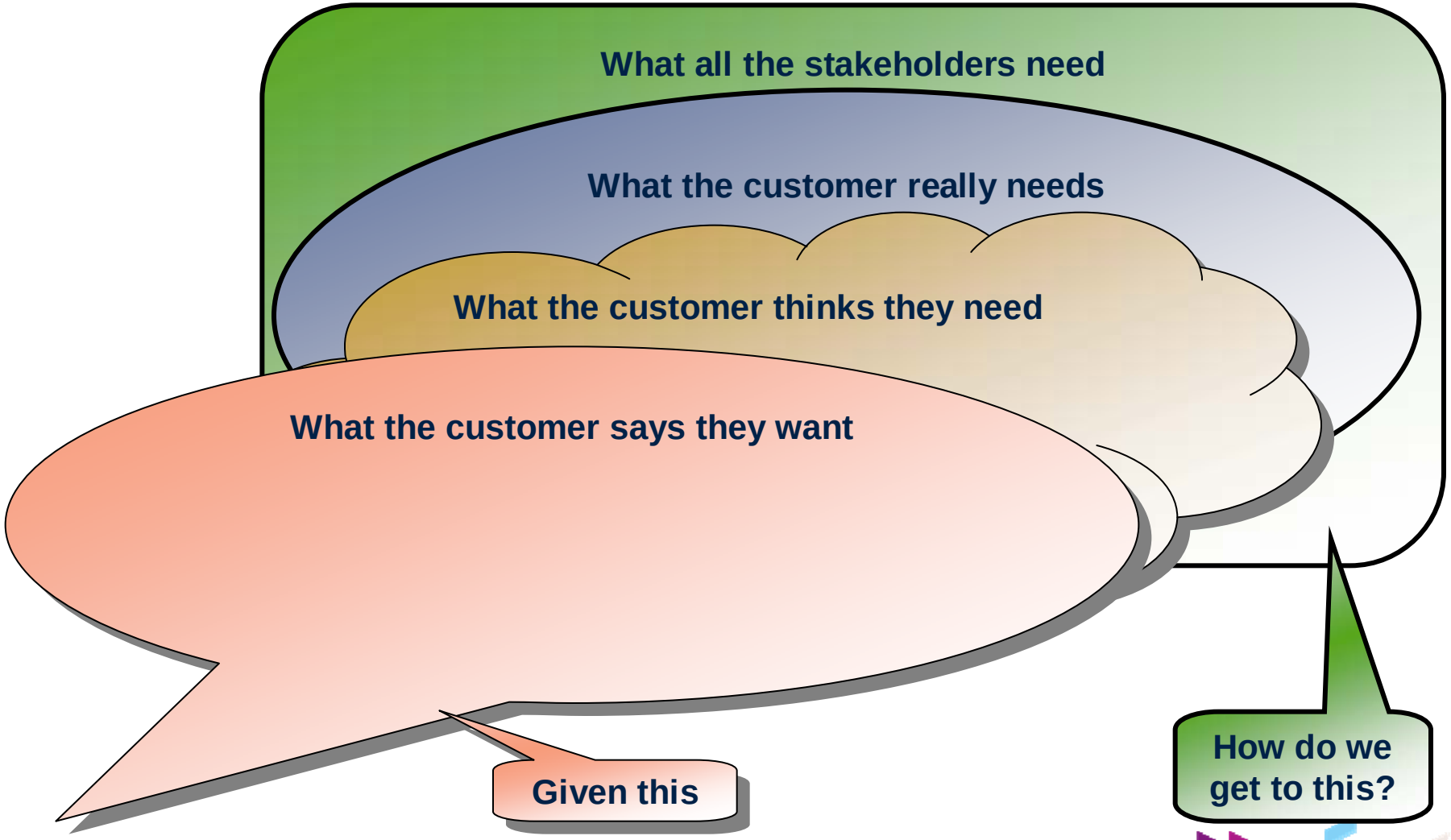
Principle 1: Recognize the needs of all stakeholders

An Exercise in clear and concise descriptive writing?

Traffic shall be able to flow at the maximum rate at all times except that in emergencies it shall be diverted away from main roads and be delayed by no more than 10 minutes, unless the emergency condition continues for more than 15 minutes in which case the delay may be increased to 30 minutes but in the event that delays are more than 10 minutes then the system shall activate an excess delay exception and shall maintain the maximum delay 10% of the stated values for a minimum of 90 minutes.

Principle 1: Recognize the needs of all stakeholders

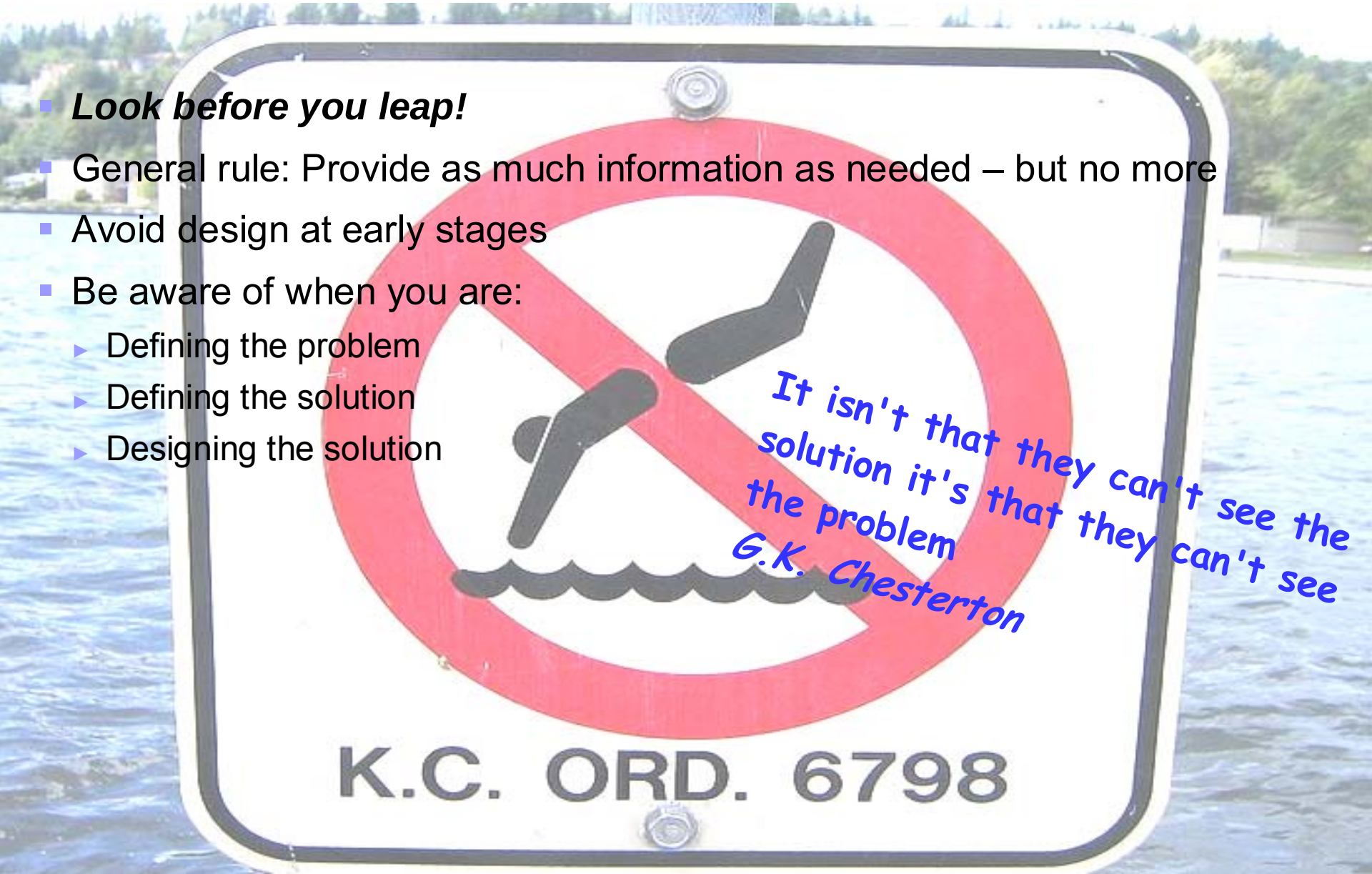
Different perspectives



Principle 2: Use abstraction to manage complexity

Keep information at the right level

- ***Look before you leap!***
- General rule: Provide as much information as needed – but no more
- Avoid design at early stages
- Be aware of when you are:
  - Defining the problem
  - Defining the solution
  - Designing the solution



It isn't that they can't see the solution it's that they can't see the problem
G.K. Chesterton

Avoid Premature Details at Top Levels

Problem

State what the stakeholders want to be able to do: **Capabilities**

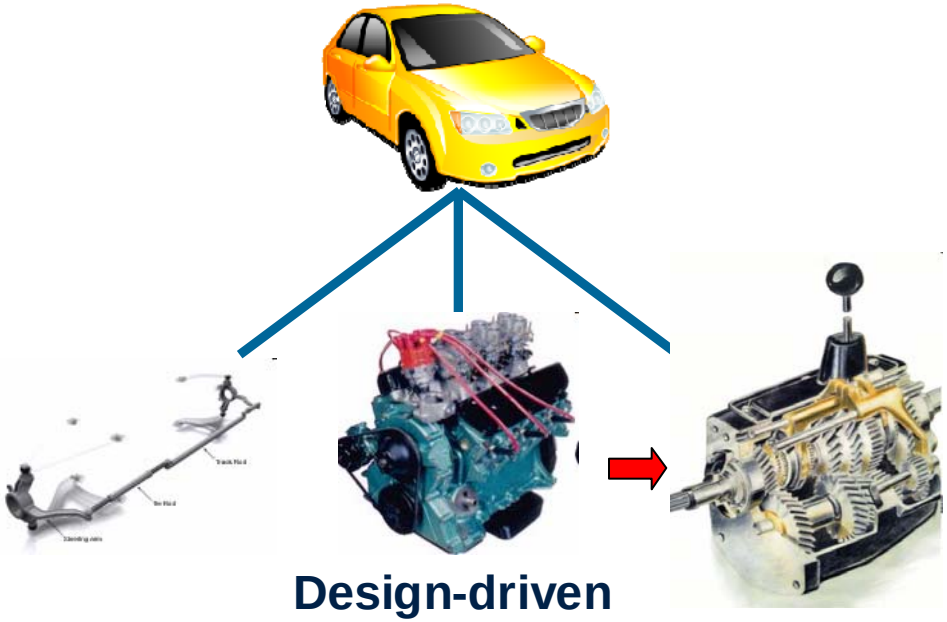
Solution

State what the system must do: **Function**

Principle 2: Use abstraction to manage complexity

Building a Requirements Hierarchy

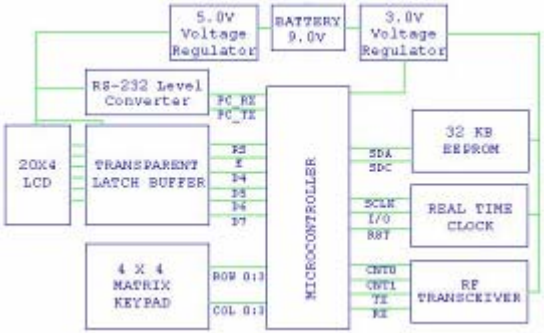
Decomposition



Design-driven

Tracing through the hierarchy

Why are we building this?



Where is this implemented?

How do I test this?

Can we show these answers? (Governance)

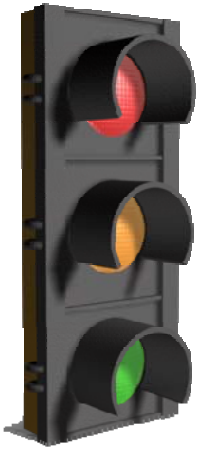
Principle 2: Use abstraction to manage complexity

Managing requirements in the hierarchy – Attributes

Identification

Type

Performance

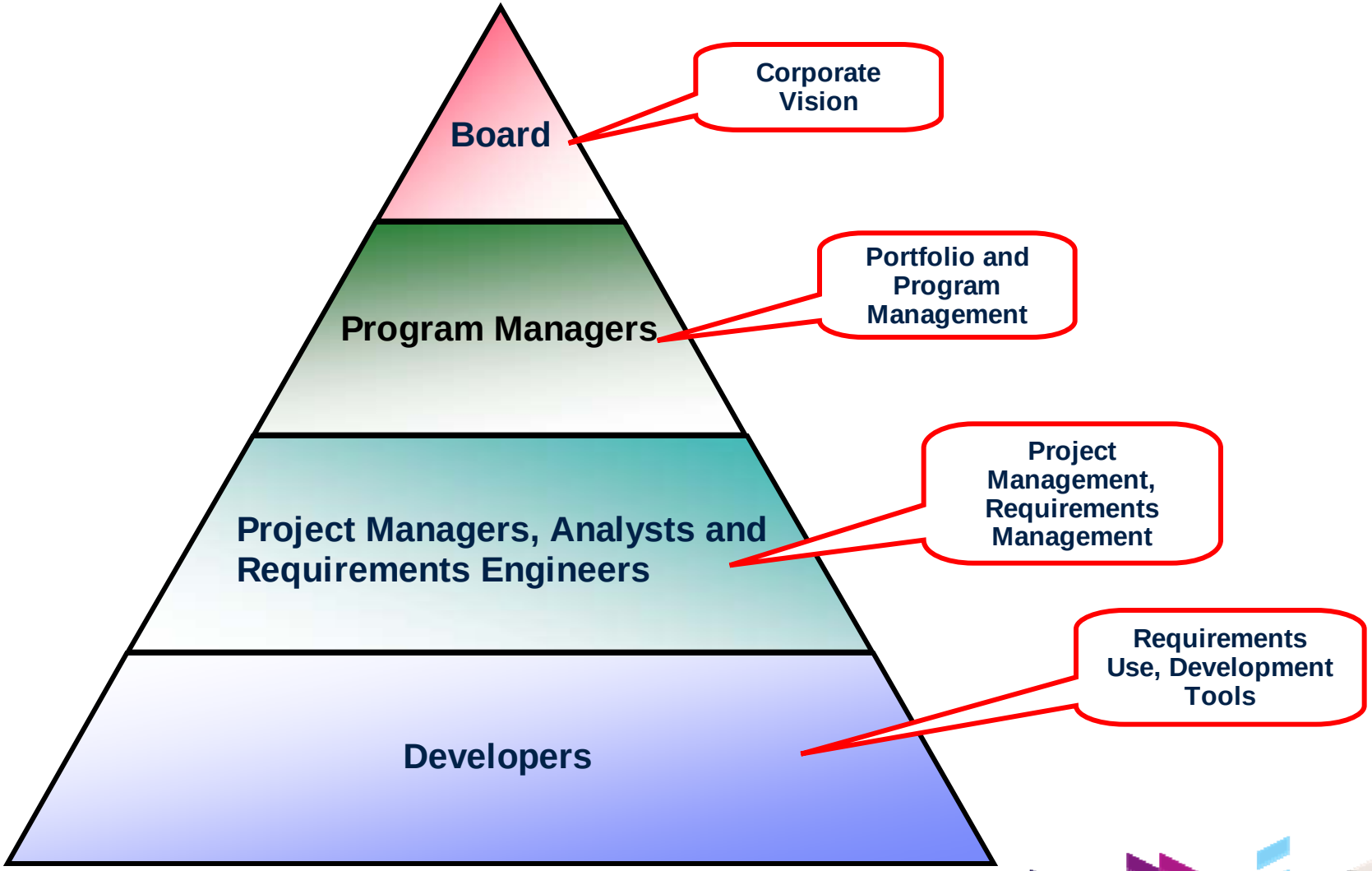


Priority

Status

Principle 3: Encourage collaboration across the lifecycle

RM across the Enterprise



Principle 3: Integrate RM across the lifecycle

Requirements Definition & Management
Must Be Integrated into the Product Lifecycle

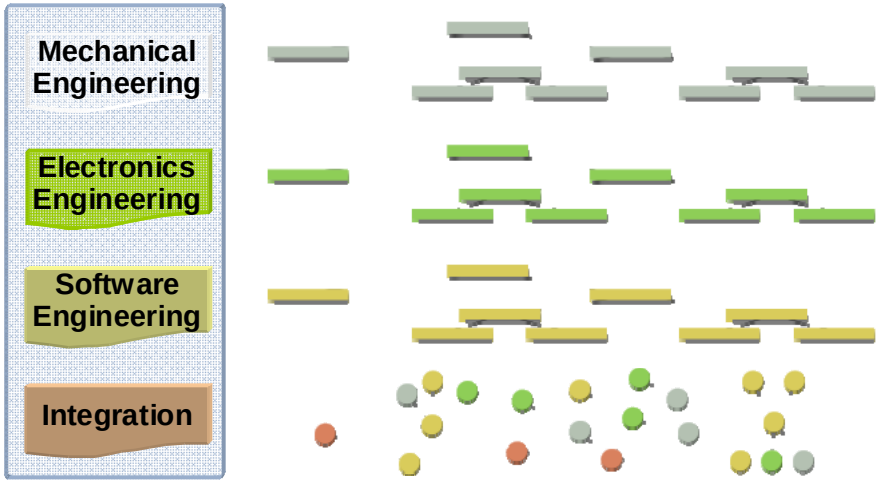
Customer Needs → Define Operation Reqt's → Define System Reqt's → Develop Concept → Preliminary Definition → Detail Definition → Product / System Build → Product / System Deliver → Run / Support / Maintain → Disposal

Business Analysis: *Enterprise Architecture, Business Process Mgmt, Product Mgmt, Portfolio Mgmt*

Program & Project Management: *Cost Accounting, Scheduling, Measurements, Reporting, Risk Mgmt*

Requirements Management

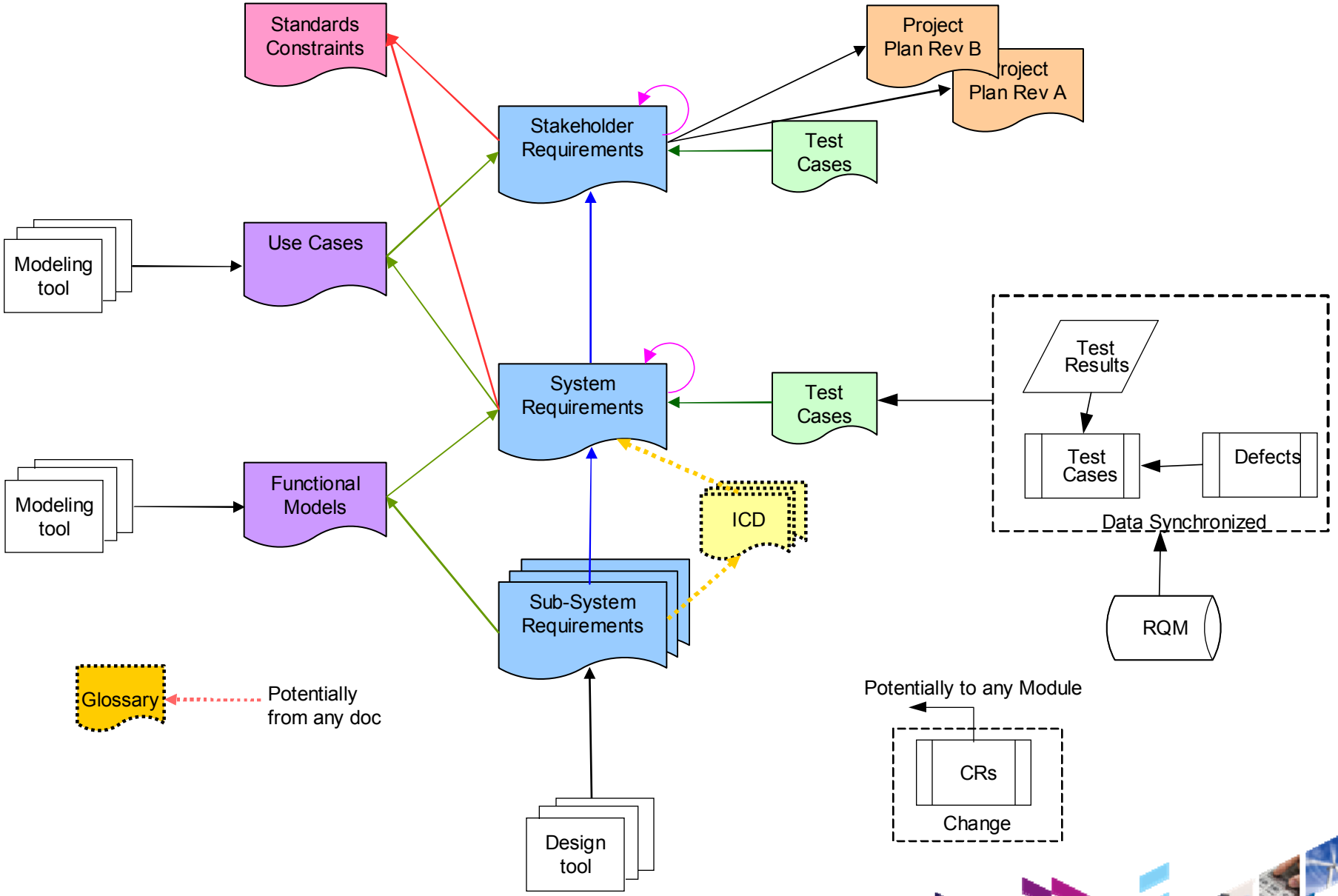
Detailed Design and Implementation



Verification and Validation – Test Management

Change and Configuration Management

Principle 3: Integrate RM across the lifecycle



Measure the requirements process

- CMMI, ITIL and other process assessment frameworks expect measurement
  - CMMI needs RM to get to level 2
  - Need measurement to understand efficiency and consistency
  - Key to continuous process improvement

Metrics Collection	Total NC	Total C	Agreed C	Delta NC	Delta C	Allocated NC	Allocated C	Proven NC	Proven C	Deleted NC	Deleted C
24 21/5/05 - 22/5/05											
24.1 21/5/05 - 21/5/05											
24.1.1 /01 - Period ending now - real time/Reqts - Standard view	4	13	0	0	0	3	10	1	4	0	0
24.1.2 /01 - Period ending now - real time/Reqts - 01 - Data Entry	4	13	0	0	0	3	10	1	4	0	0
24.2 22/5/05 - 22/5/05											
24.2.1 /01 - Period ending now - real time/Reqts - Standard view	0	0	0	0	0	0	0	0	0	0	0
24.2.2 /01 - Period ending now - real time/Reqts - 01 - Data Entry	0	0	0	0	0	0	0	0	0	0	0

Principle 4: Automate your requirements process

Effective Requirements Management realizes quantifiable savings

and with a tool you are able to measure

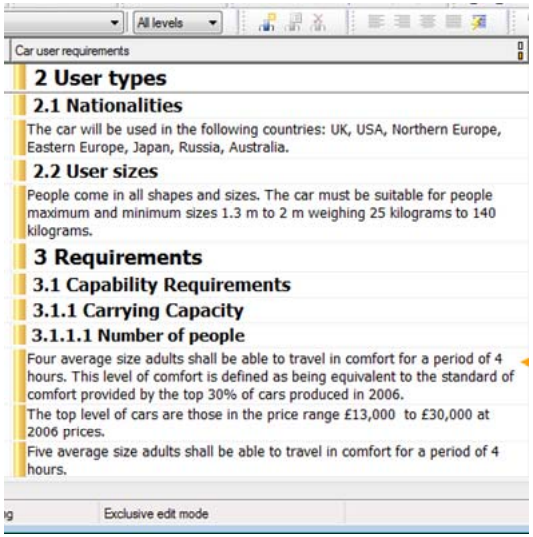
Example: how to measure and results

- Development releases consisting of typically 8000 requirements used to take 6 months
- Phase 1 - Application of robust process and tool enforcement reduced this period to 12 Weeks over a period of 1 year
- Phase 2 - Continuous process improvement for a further 12 months reduced this period to 6 weeks
- Over time, defect removal and effectiveness was 55% at phase 1, 88% at phase 2 and still improving
- Defects undetected end up with the customer – the figures represent huge improvements in cost of re-work, quality and customer satisfaction

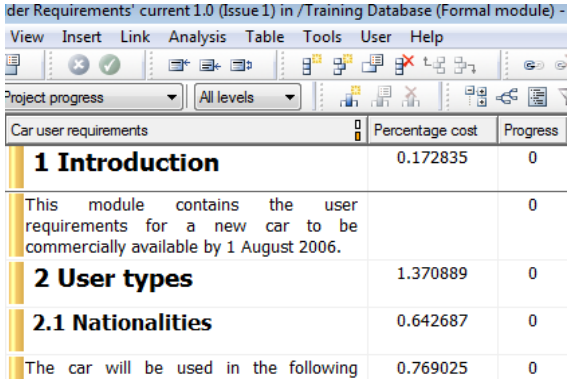
Principle 4: Automate your requirements process

Use a Requirements Management Tool

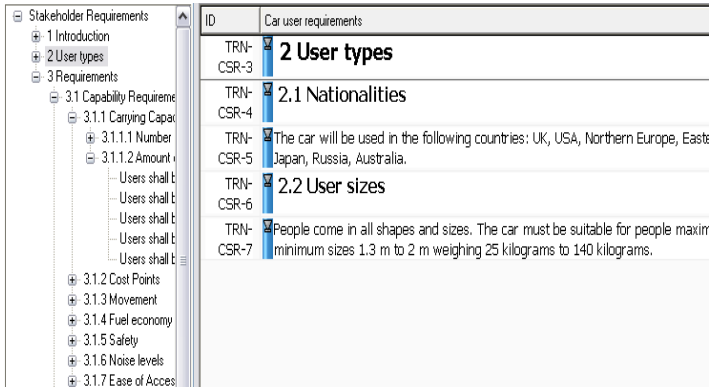
Document structure



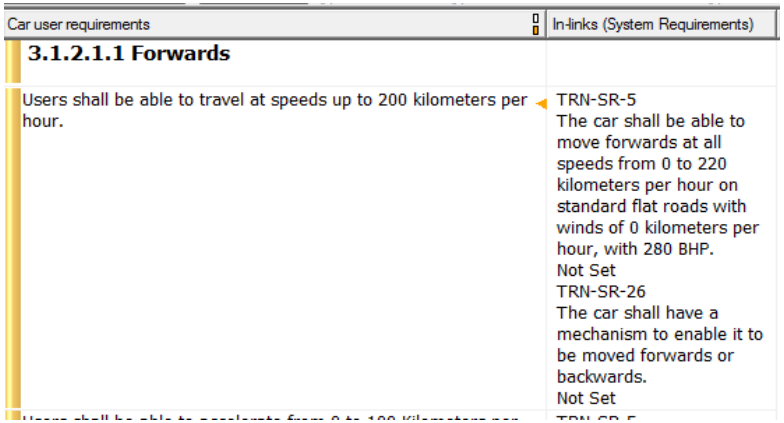
Attributes



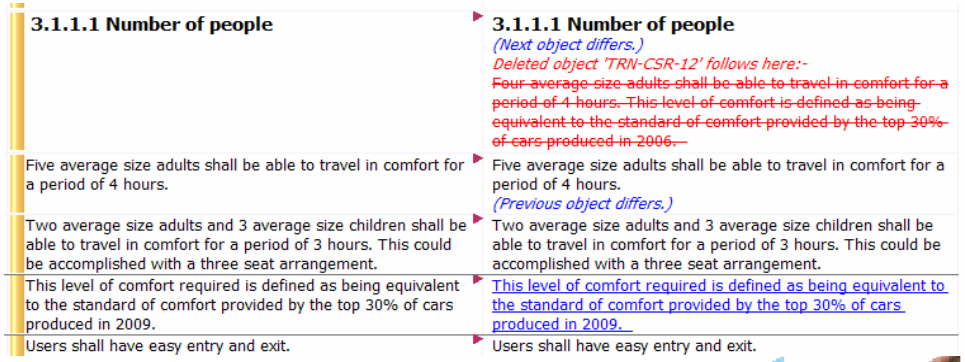
Filter to focus



View related information



View historical information



Benefits of Effective Requirements Lifecycle Management

- Greater Confidence
- Ability to manage change
- Improved customer/supplier relations
- Visibility of progress/status
- Improved Cost / Benefit Decisions

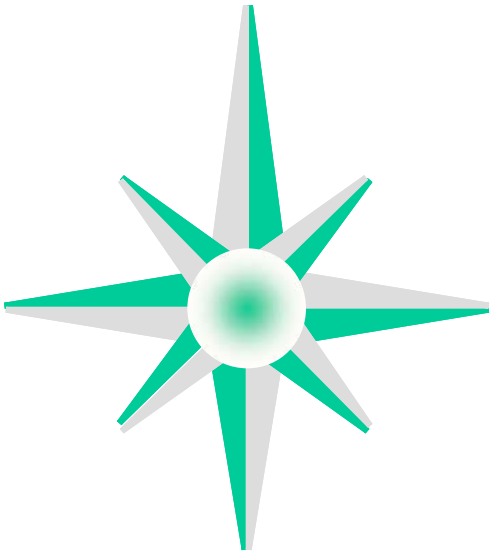
4 Principles for Effective Requirements Lifecycle Management

Recognize the needs of all stakeholders

N

Promote the use of a consistent process

W



E

Use structure to manage complexity

S

Encourage collaboration across the lifecycle

Innovate2010
The Rational Software Conference

THANK YOU