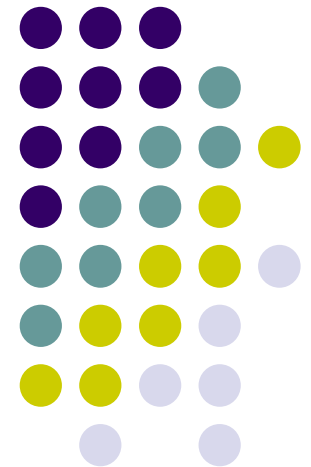


Overcoming Chaos

Evolving chaotic software systems towards flexible,
sustainable & profitable software systems.
[Requirements Engineering]

Marlize Odendaal MSc (UP)

Supervisors: Prof. Derrick Kourie
Dr. Andrew Boake





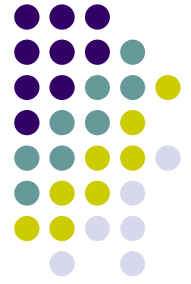
Why are we here?

Purpose of software development:

Building software systems that meet the users' requirements

Primary goals of software development teams:

- Delivering quality software
- Delivering on time
- Delivering within budget



Factors & Constraints

External Factors

- Competition
- Customers
- Compliance
- Technology
- Etc.

Internal Constraints

- Time
- Cost
- Existing systems and infrastructure
- Quality of service
- Stakeholder acceptance
- Continuity



“Our success stems from good human interactions by all participants, and our failures stem from poor human interactions”
-DeMarco & Lister

Chaos Report (Standish Group)



1995 Report

- 16.2% completed successfully
- 52.7% are challenged and overrun cost estimates by 189%
- 31.1% cancelled before completion



Success Factors

- **1. User Involvement 15.9%**
- 2. Executive Management Support 13.9%
- **3. Clear Statement of Requirements 13.0%**
- 4. Proper Planning 9.6%
- **5. Realistic Expectations 8.2%**
- 6. Smaller Project Milestones 7.7%
- 7. Competent Staff 7.2%
- 8. Ownership 5.3%
- **9. Clear Vision & Objectives 2.9%**
- 10. Hard-Working, Focused Staff 2.4%
- Other 13.9%



Challenge Factors

- 1. **Lack of User Input 12.8%**
- 2. **Incomplete Requirements & Specifications 12.3%**
- 3. **Changing Requirements & Specifications 11.8%**
- 4. Lack of Executive Support 7.5%
- 5. Technology Incompetence 7.0%
- 6. Lack of Resources 6.4%
- 7. **Unrealistic Expectations 5.9%**
- 8. **Unclear Objectives 5.3%**
- 9. Unrealistic Time Frames 4.3%
- 10. New Technology 3.7%
- Other 23.0%

Requirements...



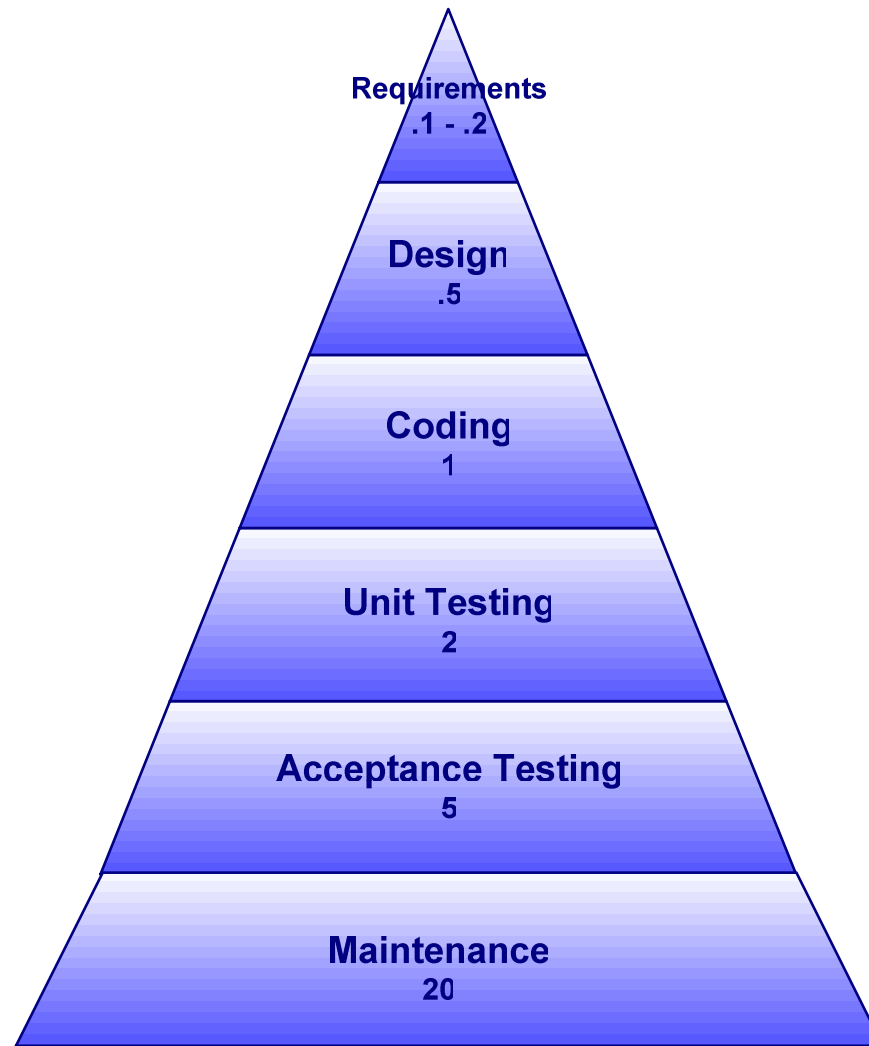
“(1) a condition or capability needed by a user to solve a problem or achieve an objective; (2) a condition or capability that must be met or possessed by a system to satisfy a contract, standard, specification or other formally imposed document.”

- IEEE standard 279



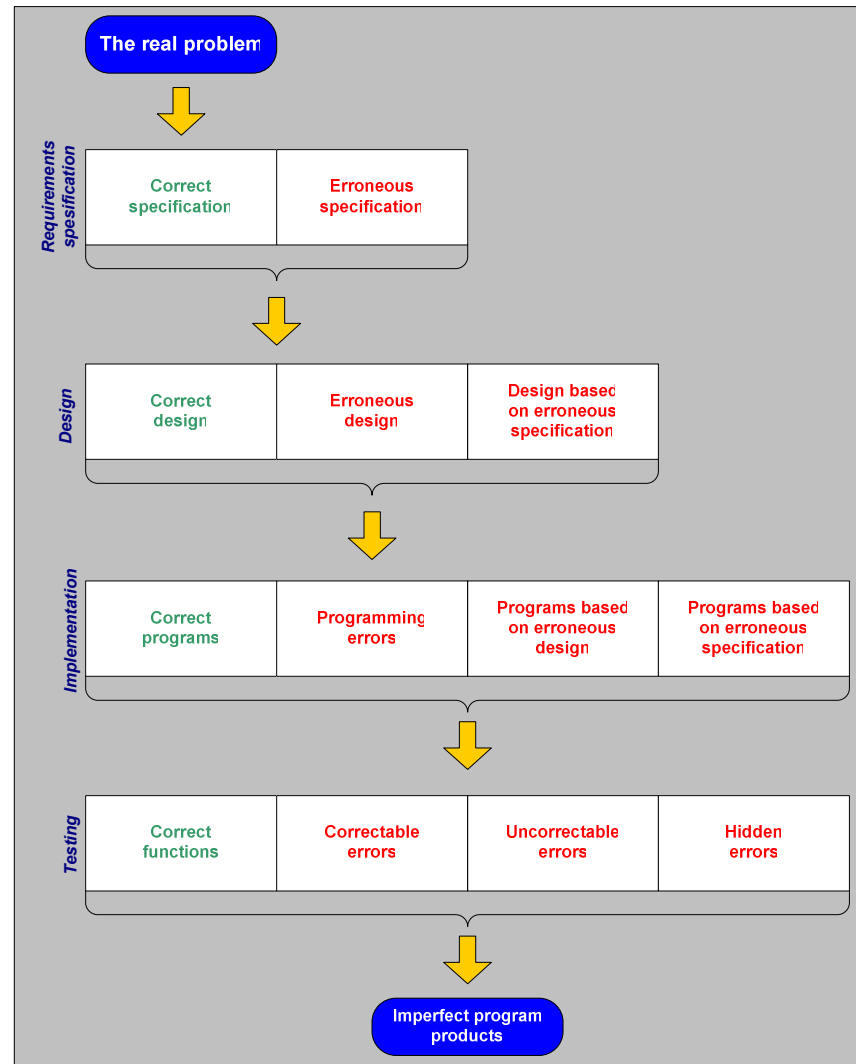
“The hardest part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all of the interfaces to people, to machines, and to other software systems. No other part of the work so *cripples the resulting system if done wrong*. No other part is more *difficult to rectify later*.”
- Brooks (1995)

Relative Defect Costs





Cumulative Effects of Errors



Requirements Engineering...



“A disciplined process-oriented approach to the definition, documentation and maintenance of software requirements throughout the software development life cycle.”

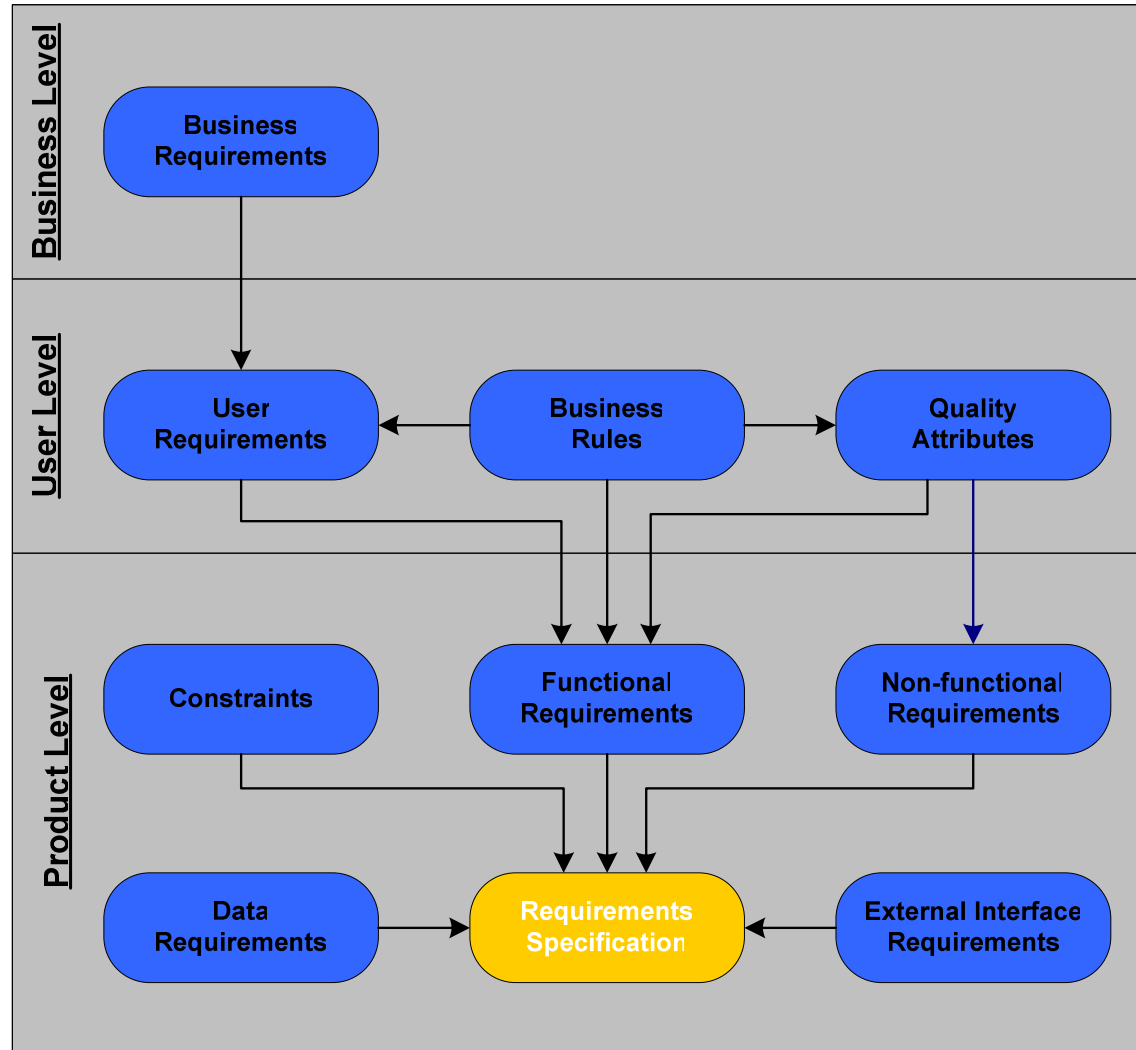
- Westfall

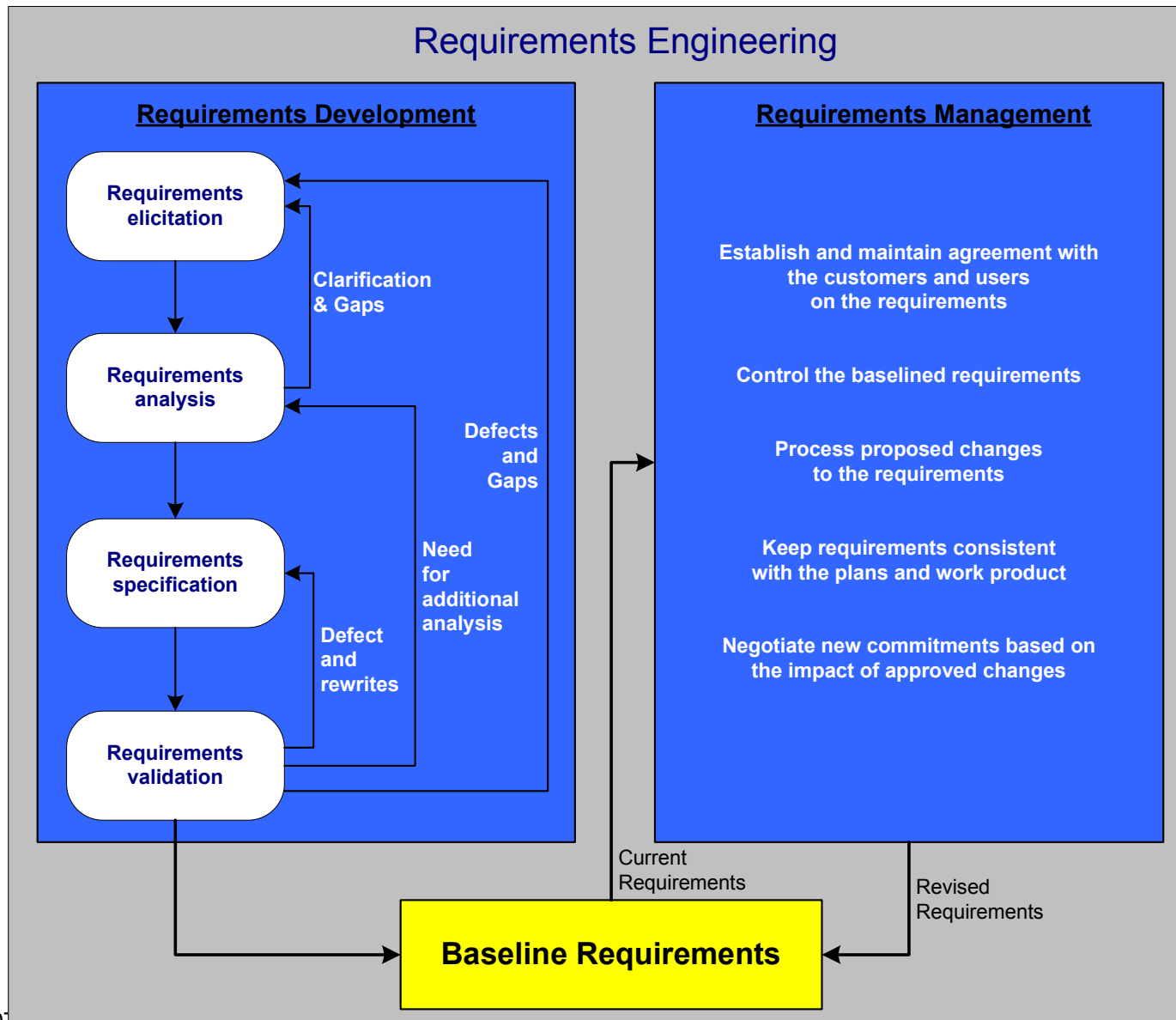
“A systematic approach to eliciting, organizing and documenting the requirements of the system, and a process that establishes and maintains agreement between the customer and the project team on the changing requirements of the system.”

- Leffingwell and Widrig



Types of Requirements





Baseline Requirements



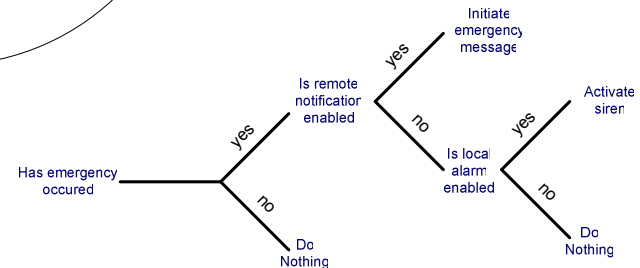
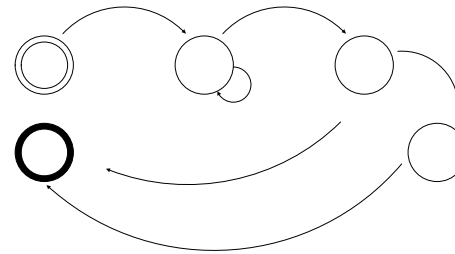
Requirements Elicitation

- Interviewing & Questionnaires
- Requirements Workshops
- Brainstorming & Idea reduction
- Storyboarding
- Use Cases
- Role playing
- Prototyping
- Documentation Studies

Requirement Specification Methods

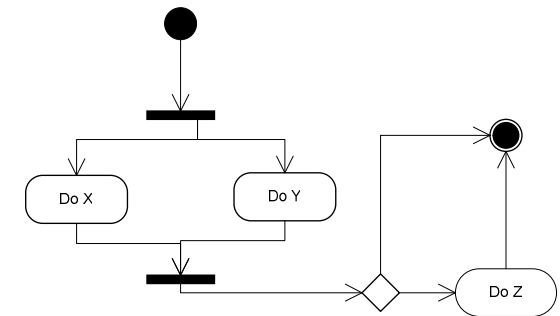
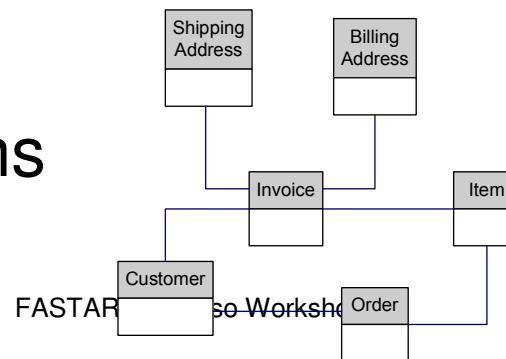


- Pseudocode
- Finite State Machines
- Decision Trees & Decision Tables
- Activity Diagrams
- Entity Relationship Models
- Object Oriented Modeling
- Data Flow Diagrams



```

Set SUM(x) = 0
FOR each customer x
  IF customer purchase paid support
    AND ((Current month) >= (2 months after ship date))
    AND ((Current month) <= (14 months after ship date))
  THEN Sum(x)= Sum(x) + (amount customer paid) / 2
  
```





Ensuring Correctness

Quality Measures

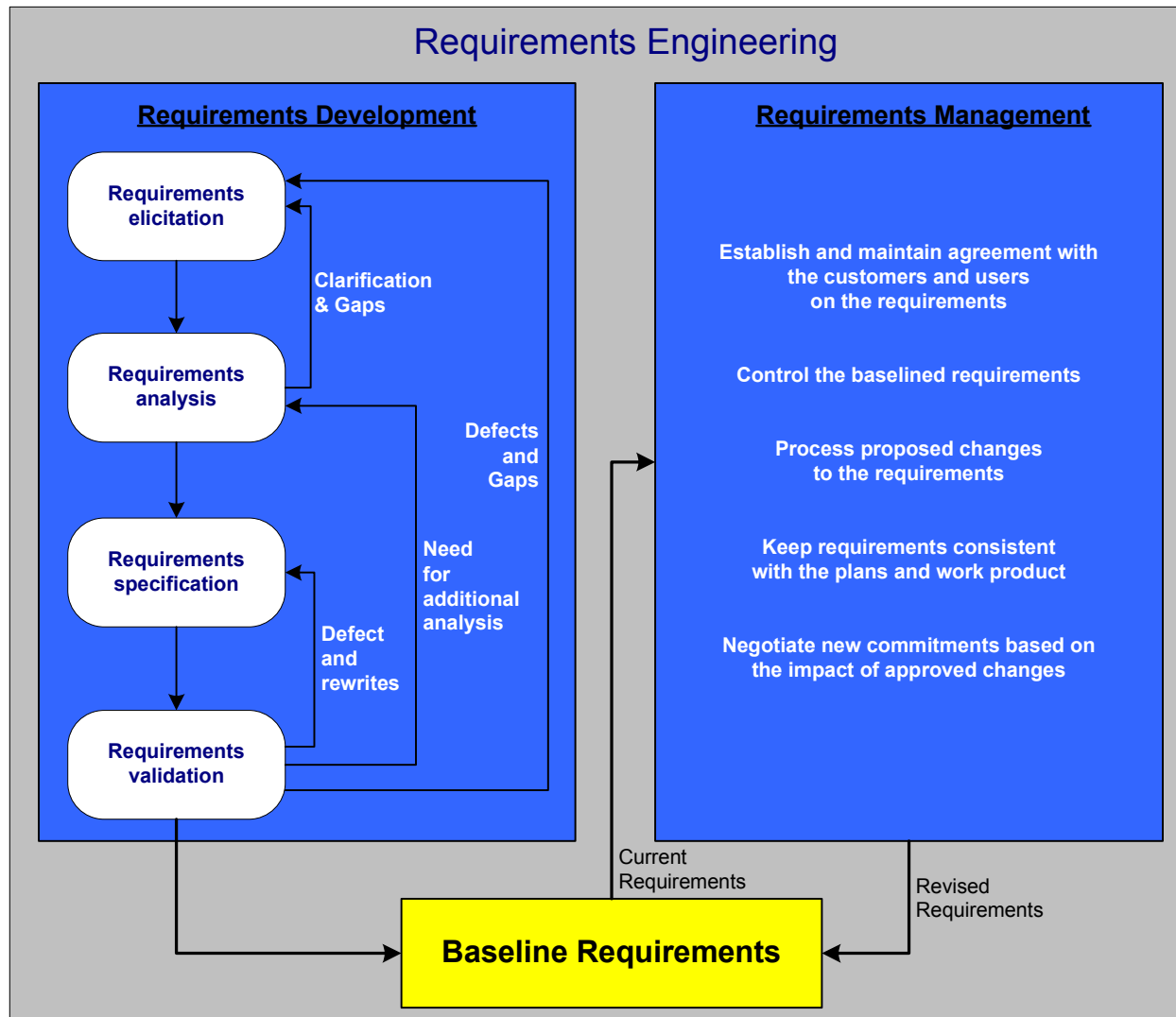
- Correct
- Unambiguous
- Concise
- Finite
- Measurable
- Feasible
- Testable
- Traceable

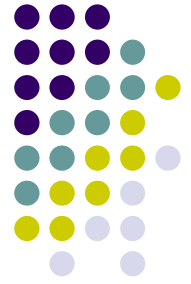
Validation

- Complete
- Consistent
- Modifiable
- Annotated
- Understandable

Methods

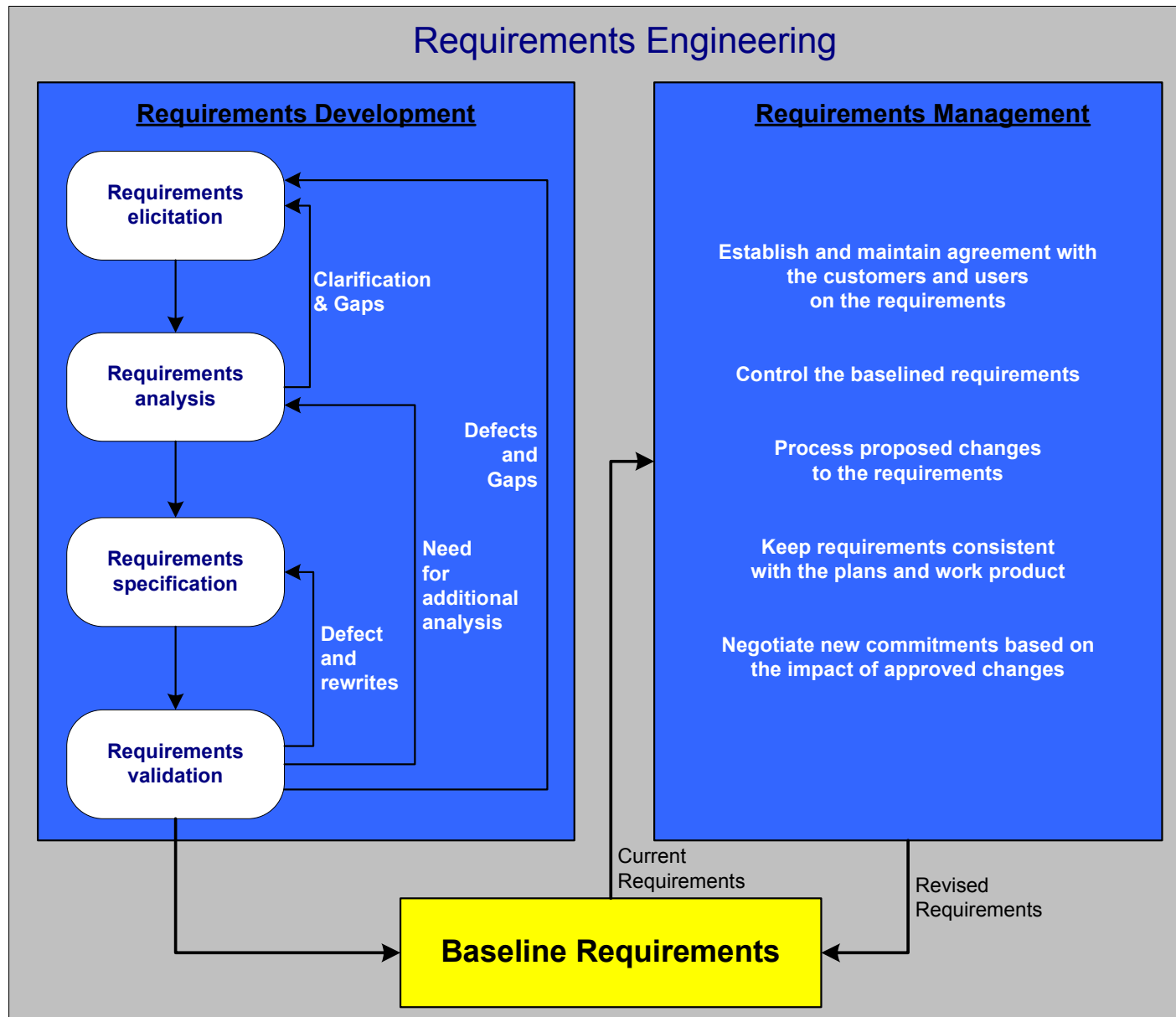
- Peer reviews
- Test Cases
- Tools?





Requirements Baseline

ID	Feature	Priority	Effort	Risk	Comments
3	Feature 3	Critical	Med	Med	
5	Feature 5	Critical	High	High	Mitigate risk by...
7	Feature 7	Important	low	Low	
<i>V1.0 Mandatory Baseline: Everything above must be included or release will be delayed</i>					
1	Feature 1	Important	High	Med	
2	Feature 2	Useful	Med	High	
<i>V1.0 Optional: Do as many of the preceding as possible</i>					
4	Feature 4	Useful	Med	Med	
6	Feature 6	Useful	Med	Low	



Case Study



Discovery Life Group Risk Systems

Current Business Situation (Jan 07)



- R275M API (5% of market share)
- 1000 schemes with 88 000 members
- Double the operating cost compared to competitors (18% of API)
- Customer satisfaction at 72%
- Lose 25% of new business at quotes stage due to absence of investment products (Retirement Funds or Pension)



Current Situation

- Current system has 54% functionality
- 10 Projects and plenty CRs to close gap
- Duration of 30 to 36 months with 18 staff
- Parts of systems to be re-written
- Limited capacity on new products



Business Objectives

- Increase API to R550M in 3 years – R1B in 5 years
- Decrease the operating cost to 9% of API
- New pricing strategy (Integration to other products)
- Increase customer satisfaction to 90%
- Shorter time to market for new products