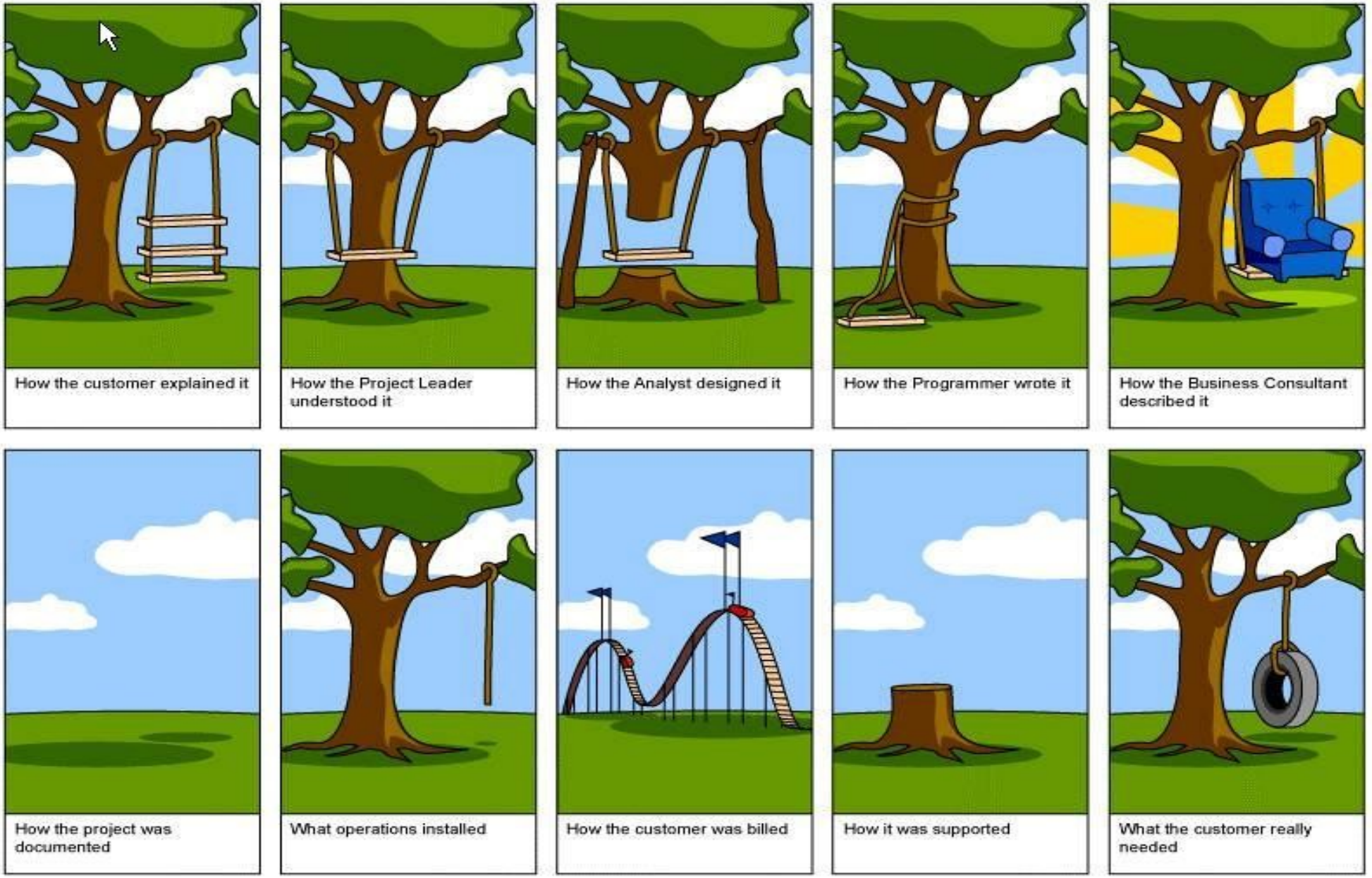
Processes in Software Development

Presented 11.3.2008 by Lars Yde, M.Sc., at ”Selected Topics in Software Development”, DIKU spring semester 2008



Software "hall of shame"

YEAR	COMPANY	OUTCOME (COSTS IN US $)
2005	Hudson Bay Co. [Canada]	Problems with inventory system contribute to $33.3 million* loss.
2004–05	UK Inland Revenue	Software errors contribute to $3.45 billion* tax-credit overpayment.
2004	Avis Europe PLC [UK]	Enterprise resource planning (ERP) system canceled after $54.5 million† is spent.
2004	Ford Motor Co.	Purchasing system abandoned after deployment costing approximately $400 million.
2004	J Sainsbury PLC [UK]	Supply-chain management system abandoned after deployment costing $527 million.†
2004	Hewlett-Packard Co.	Problems with ERP system contribute to $160 million loss.
2003–04	AT&T Wireless	Customer relations management (CRM) upgrade problems lead to revenue loss of $100 million.
2002	McDonald's Corp.	The Innovate information-purchasing system canceled after $170 million is spent.
2002	Sydney Water Corp. [Australia]	Billing system canceled after $33.2 million† is spent.
2002	CIGNA Corp.	Problems with CRM system contribute to $445 million loss.
2001	Nike Inc.	Problems with supply-chain management system contribute to $100 million loss.
2001	Kmart Corp.	Supply-chain management system canceled after $130 million is spent.
2000	Washington, D.C.	City payroll system abandoned after deployment costing $25 million.
1999	United Way	Administrative processing system canceled after $12 million is spent.
1999	State of Mississippi	Tax system canceled after $11.2 million is spent; state receives $185 million damages.
1999	Hershey Foods Corp.	Problems with ERP system contribute to $151 million loss.
1998	Snap-on Inc.	Problems with order-entry system contribute to revenue loss of $50 million.
1997	U.S. Internal Revenue Service	Tax modernization effort canceled after $4 billion is spent.
1997	State of Washington	Department of Motor Vehicle (DMV) system canceled after $40 million is spent.
1997	Oxford Health Plans Inc.	Billing and claims system problems contribute to quarterly loss; stock plummets, leading to $3.4 billion loss in corporate value.
1996	Arianespace [France]	Software specification and design errors cause $350 million Ariane 5 rocket to explode.
1996	FoxMeyer Drug Co.	$40 million ERP system abandoned after deployment, forcing company into bankruptcy.
1995	Toronto Stock Exchange [Canada]	Electronic trading system canceled after $25.5 million** is spent.
1994	U.S. Federal Aviation Administration	Advanced Automation System canceled after $2.6 billion is spent.
1994	State of California	DMV system canceled after $44 million is spent.
1994	Chemical Bank	Software error causes a total of $15 million to be deducted from 100 000 customer accounts.
1993	London Stock Exchange [UK]	Taurus stock settlement system canceled after $600 million** is spent.
1993	Allstate Insurance Co.	Office automation system abandoned after deployment, costing $130 million.
1993	London Ambulance Service [UK]	Dispatch system canceled in 1990 at $11.25 million**; second attempt abandoned after deployment, costing $15 million.**
1993	Greyhound Lines Inc.	Bus reservation system crashes repeatedly upon introduction, contributing to revenue loss of $61 million.
1992	Budget Rent-A-Car, Hilton Hotels, Marriott International, and AMR [American Airlines]	Travel reservation system canceled after $165 million is spent.

SOURCES: *BUSINESS WEEK, CEO MAGAZINE, COMPUTERWORLD, INFOWEEK, FORTUNE, THE NEW YORK TIMES, TIME,* AND *THE WALL STREET JOURNAL*

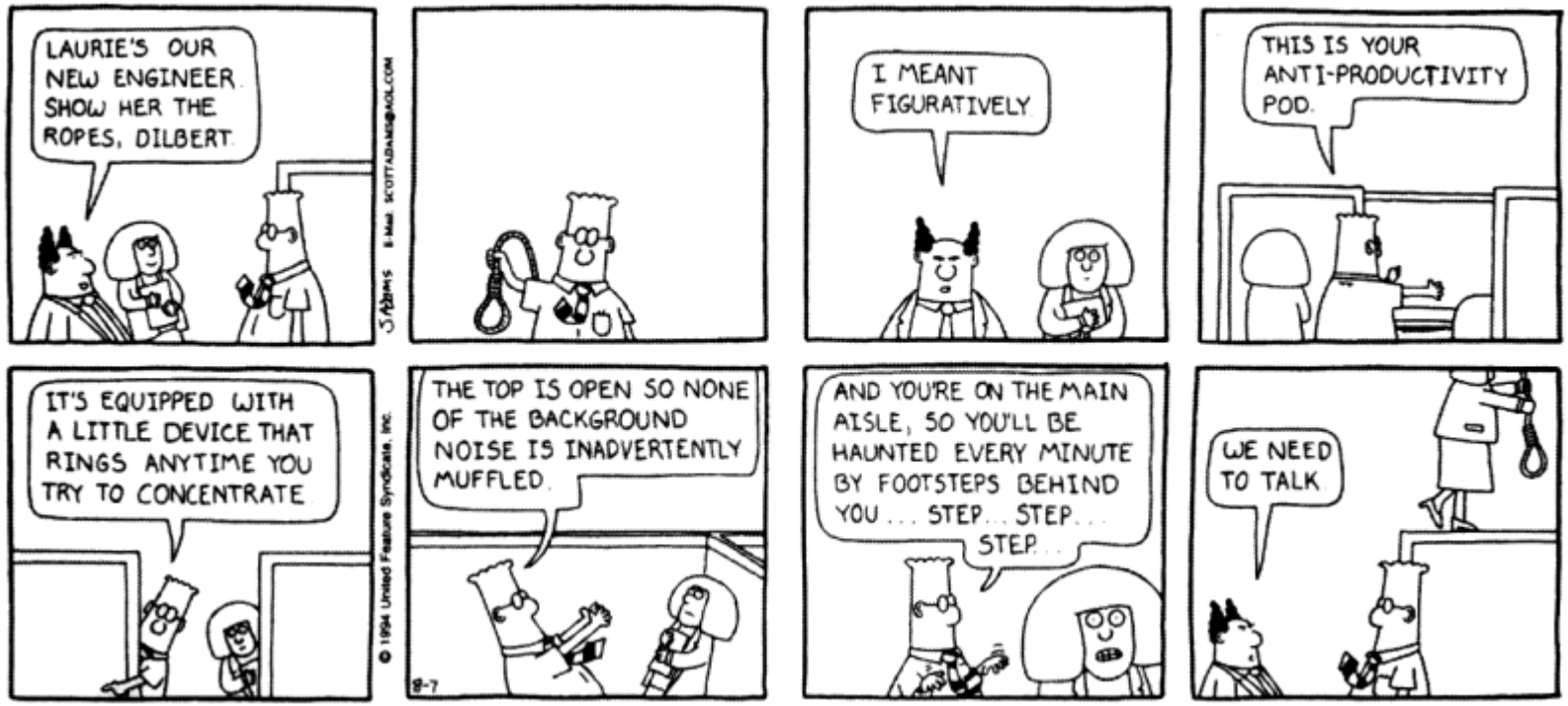
Classic mistakes

People-Related Mistakes	Process-Related Mistakes	Product-Related Mistakes	Technology-Related Mistakes
1. Undermined motivation	14. Overly optimistic schedules	28. Requirements gold-plating	33. Silver-bullet syndrome
2. Weak personnel	16. Insufficient risk management	29. Feature creep	34. Overestimated savings from new tools or methods
3. Uncontrolled problem employees	17. Contractor failure Insufficient planning	30. Developer gold-plating	35. Switching tools in the middle of a project
4. Heroics	18. Abandonment of planning under pressure	31. Push me, pull me negotiation	36. Lack of automated source-code control
5. Adding people to a late project	19. Wasted time during the fuzzy front end	32. Research-oriented development	
6. Noisy, crowded offices	20. Shortchanged upstream activities		
7. Friction between developers and customers	21. Inadequate design		
8. Unrealistic expectations	22. Shortchanged quality assurance		
9. Lack of effective project sponsorship	23. Insufficient management controls		
10. Lack of stakeholder buy-in	24. Premature or too frequent convergence		
11. Lack of user input	25. Omitting necessary tasks from estimates		
12. Politics placed over substance	26. Planning to catch up later		
13. Wishful thinking	27. Code-like-hell programming		

ACM Code of Ethics

As an ACM member I will

- Contribute to society and human well-being.
- Avoid harm to others.
- Be honest and trustworthy.
- Be fair and take action not to discriminate.
- Honor property rights including copyrights and patent.
- Give proper credit for intellectual property.
- Respect the privacy of others.
- Honor confidentiality.

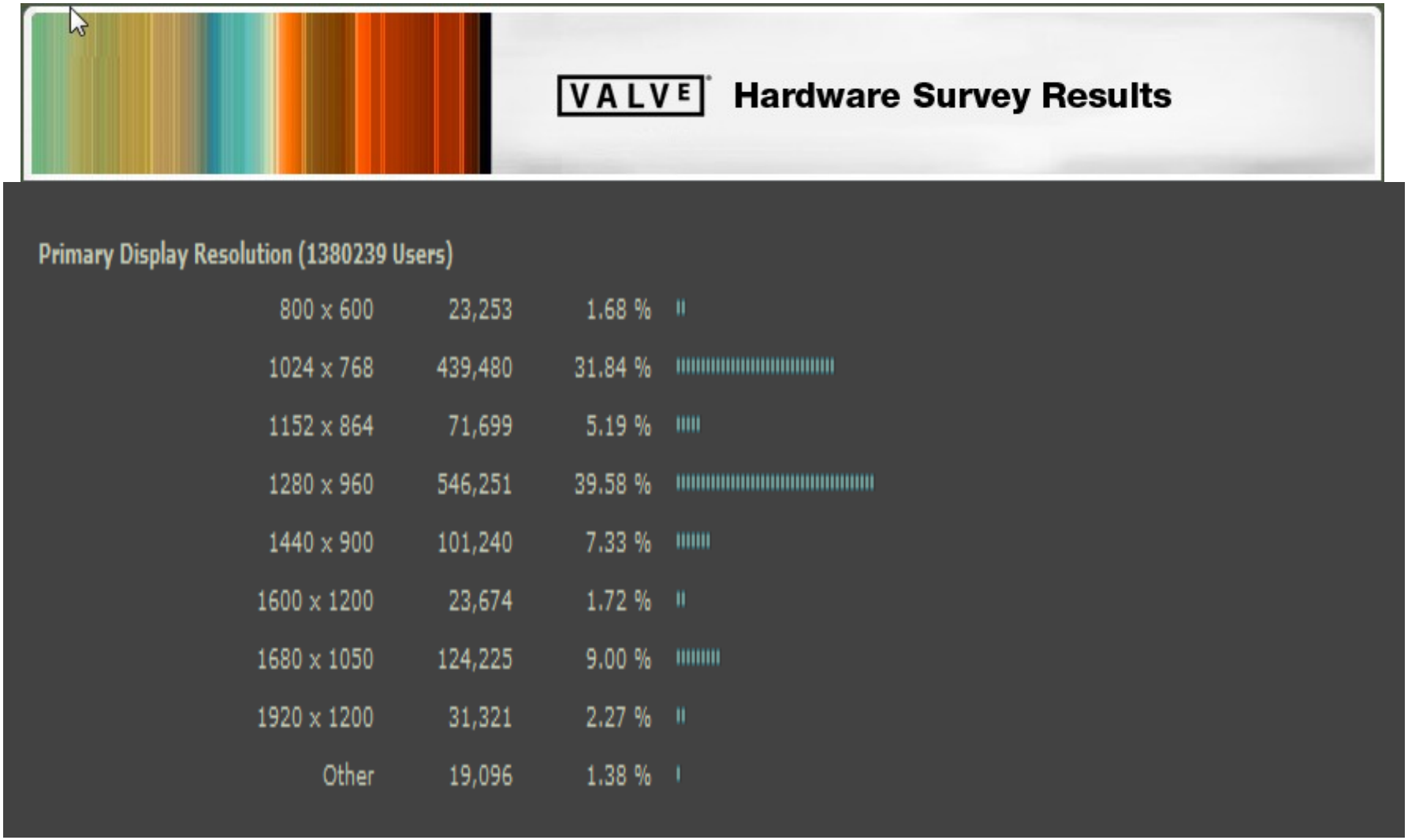


Catastrophic bugs

July 28, 1962 -- Mariner I space probe.
1982 -- Soviet gas pipeline.
1985-1987 -- Therac-25 medical accelerator.
1988 -- Buffer overflow in Berkeley Unix finger daemon.
1988-1996 -- Kerberos Random Number Generator.
January 15, 1990 -- AT&T Network Outage.
1993 -- Intel Pentium floating point divide.
1995/1996 -- The Ping of Death.
June 4, 1996 -- Ariane 5 Flight 501.
November 2000 -- National Cancer Institute, Panama City.

What users do, not what they say

Source: steampowered.com



The 10th edition of the annual CHAOS report from The Standish Group

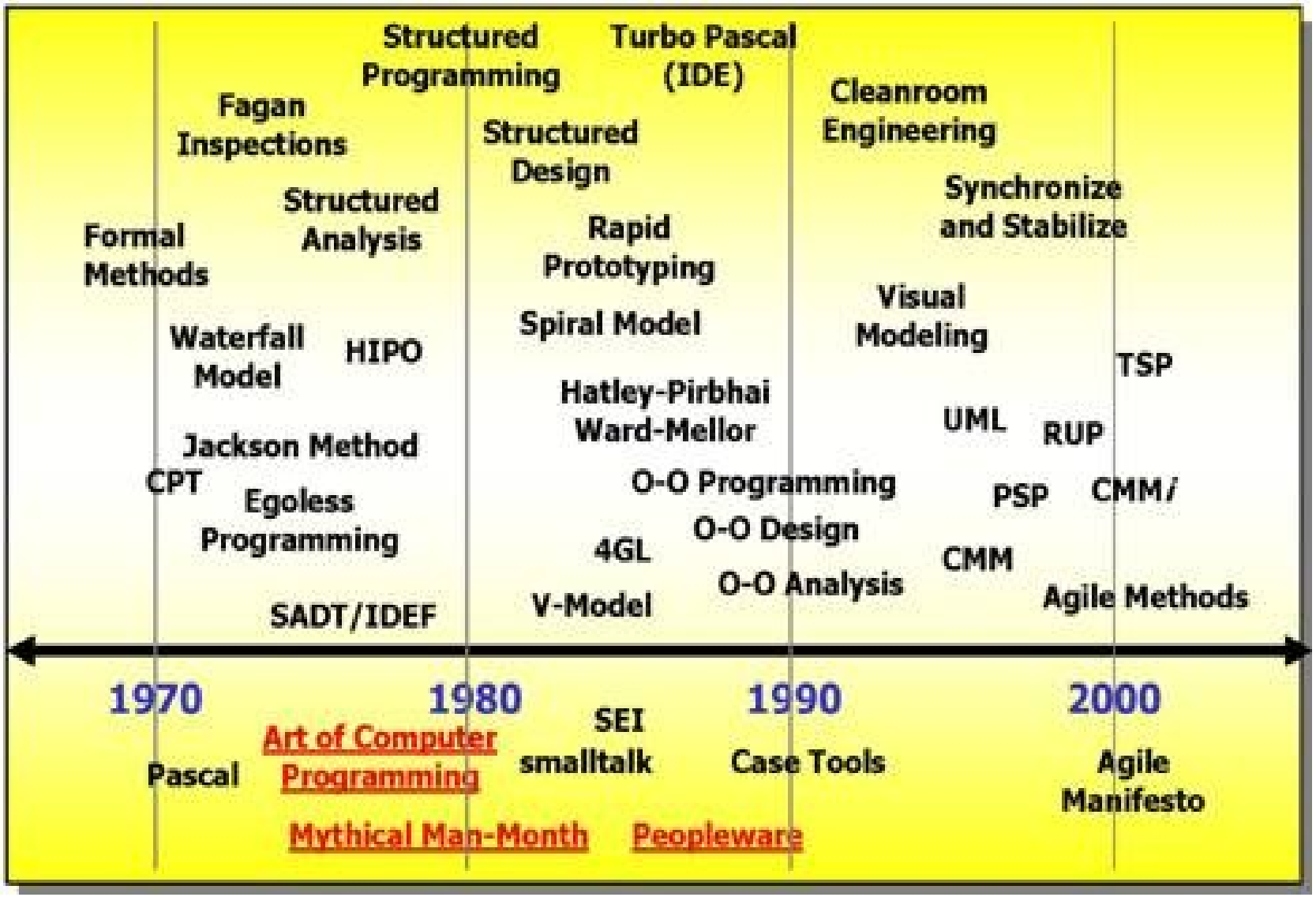
”The 2004 CHAOS report, entitled “CHAOS Chronicles,” found total U.S. project waste to be **$55 billion**, made up of **$38 billion** in lost dollar value and **$17 billion in cost overruns**. Total project spending was found to be $255 billion in the 2004 report.

In 1994, The Standish Group estimated U.S. IT projects wasted **$140 billion—$80 billion** of that from **failed projects**—out of a total of **$250 billion** in project spending.”

Standish Chairman Jim Johnson says, “The primary reason is the projects have gotten a lot smaller. Doing projects with iterative processing as opposed to the waterfall method, which called for all project requirements to be defined up front, is a major step forward.”

Softwaremag.com, January Issue 2004

Brief history of software dev meth

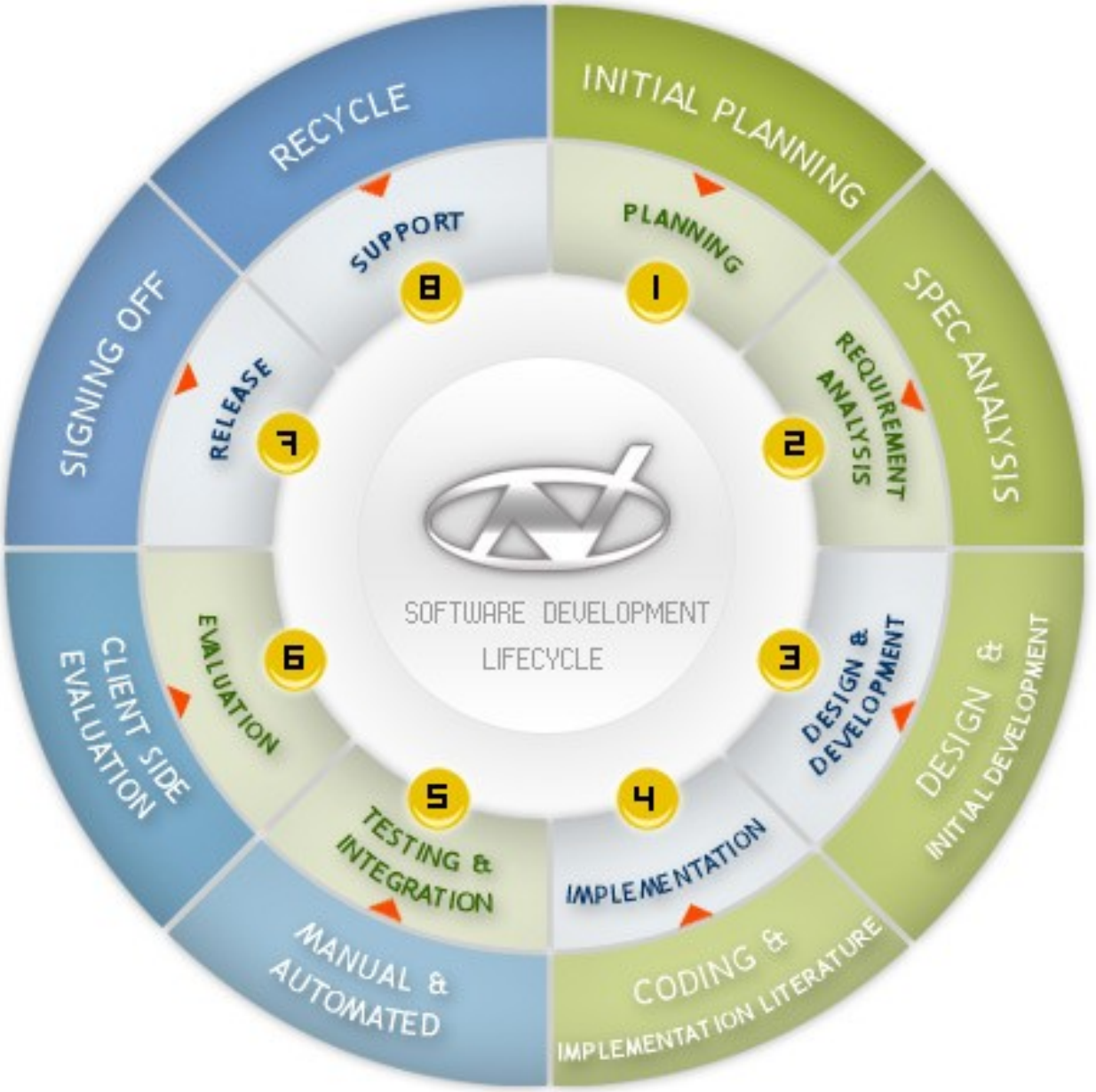


Scripted methodology / framework

- Waterfall / Big Design Up Front (BDUF)
- Spiral
- Unified Process
- Microsoft Solutions Framework
- CMM / CMMI
- ITIL
- Six Sigma
- Et al

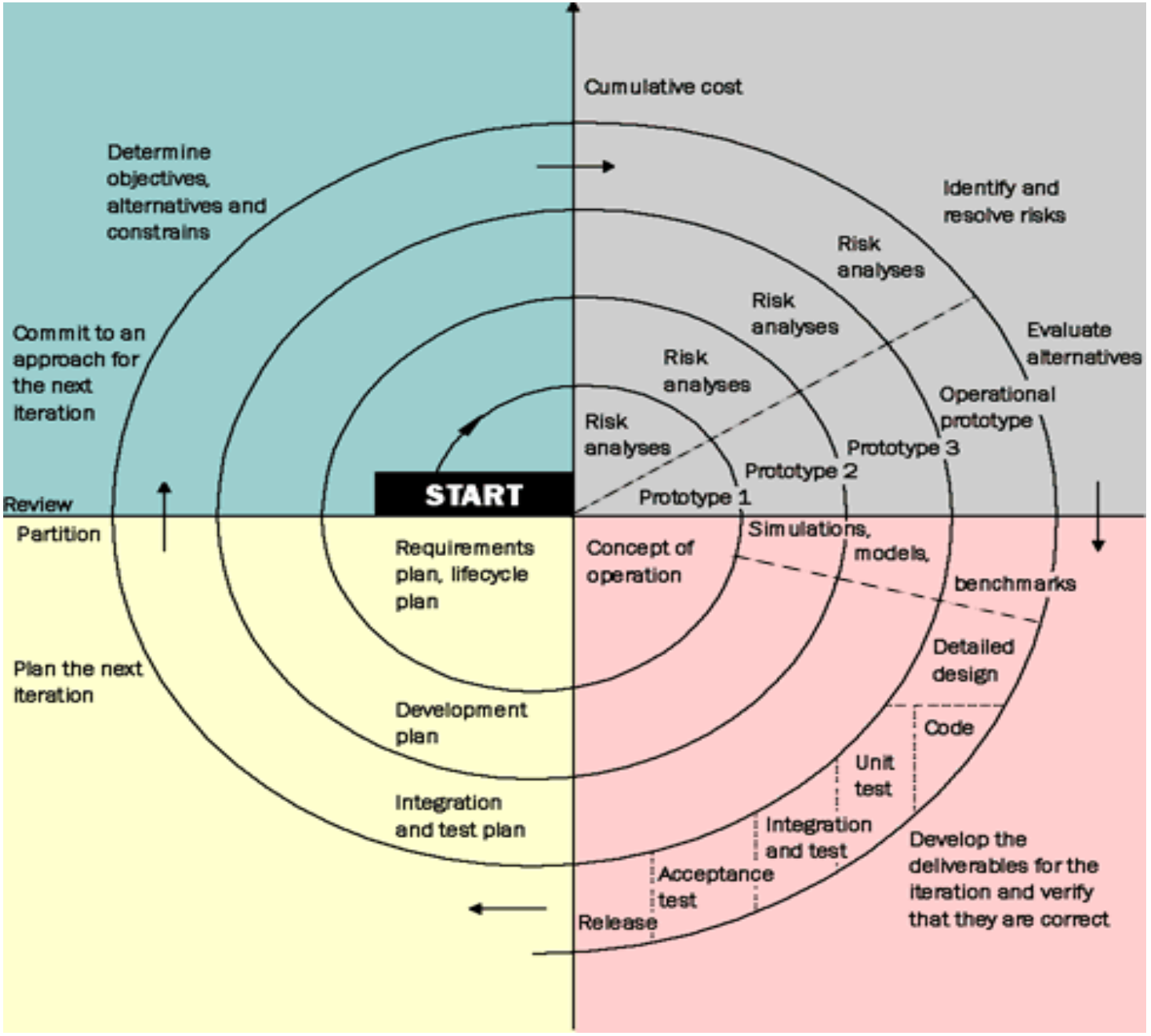
The virtuous cycle that never was

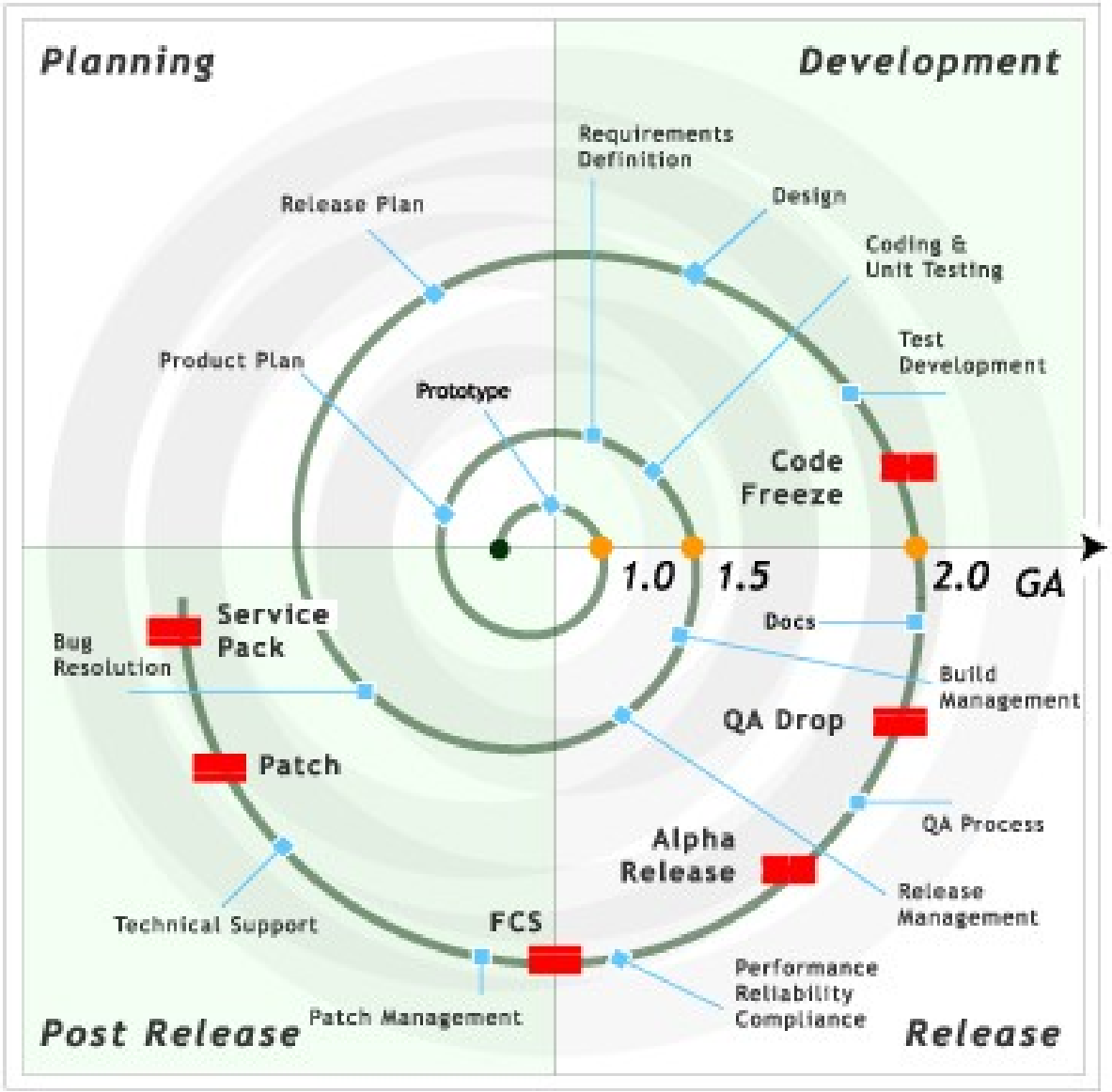
Source: notetech.com



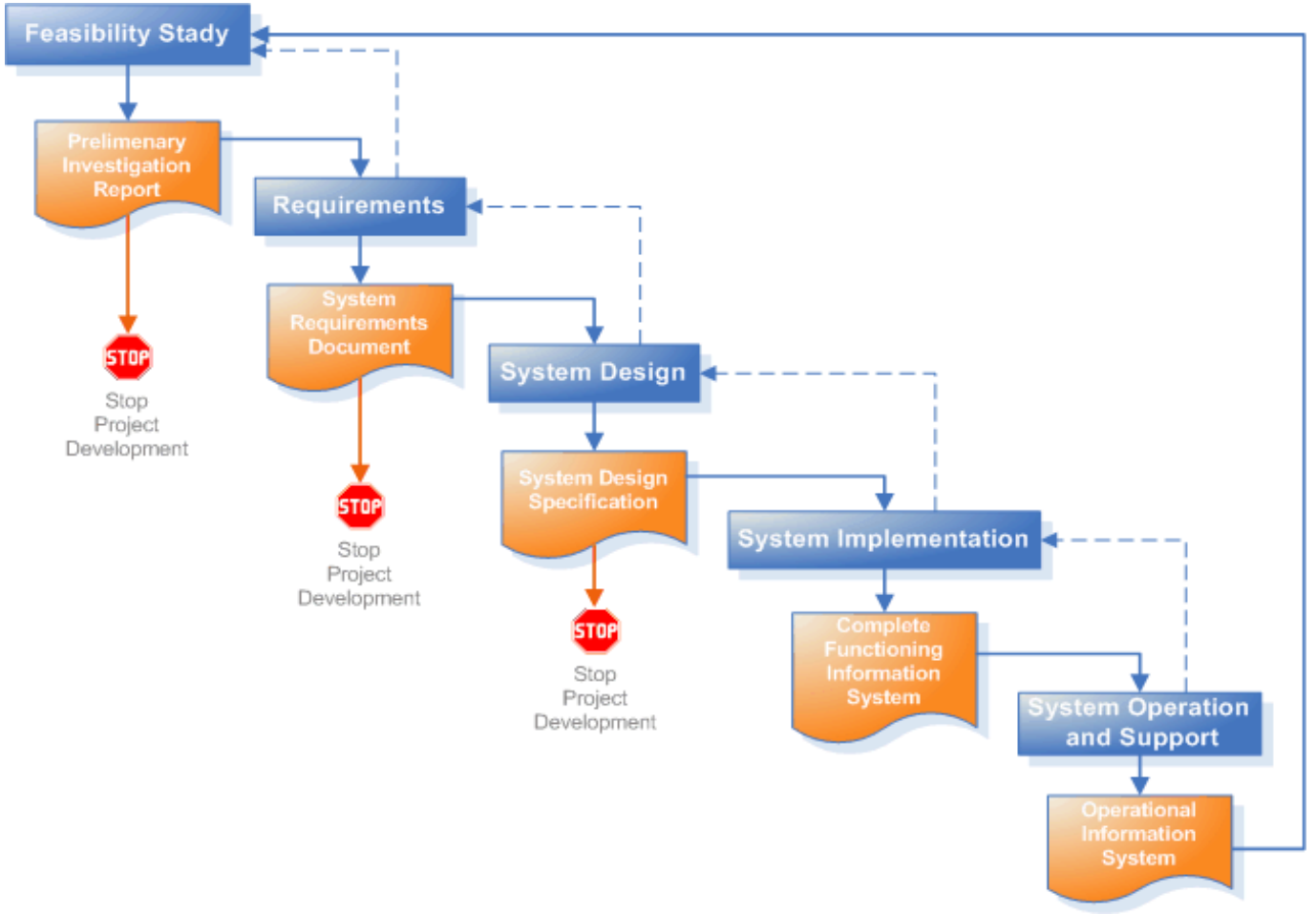
Scripted methodology #2: spiral

Boehm, Barry: ”A Spiral Model of Software Development and Enhancement”, 1988




Planning
Development
Requirements Definition
Release Plan
Design
Coding & Unit Testing
Product Plan
Prototype
Test Development
Code Freeze
1.0
1.5
2.0
GA
Service Pack
Bug Resolution
Docs
Build Management
QA Drop
Patch
QA Process
Alpha Release
Release Management
Technical Support
FCS
Performance Reliability Compliance
Patch Management
Post Release
Release

Scripted methodology #1 : waterfall

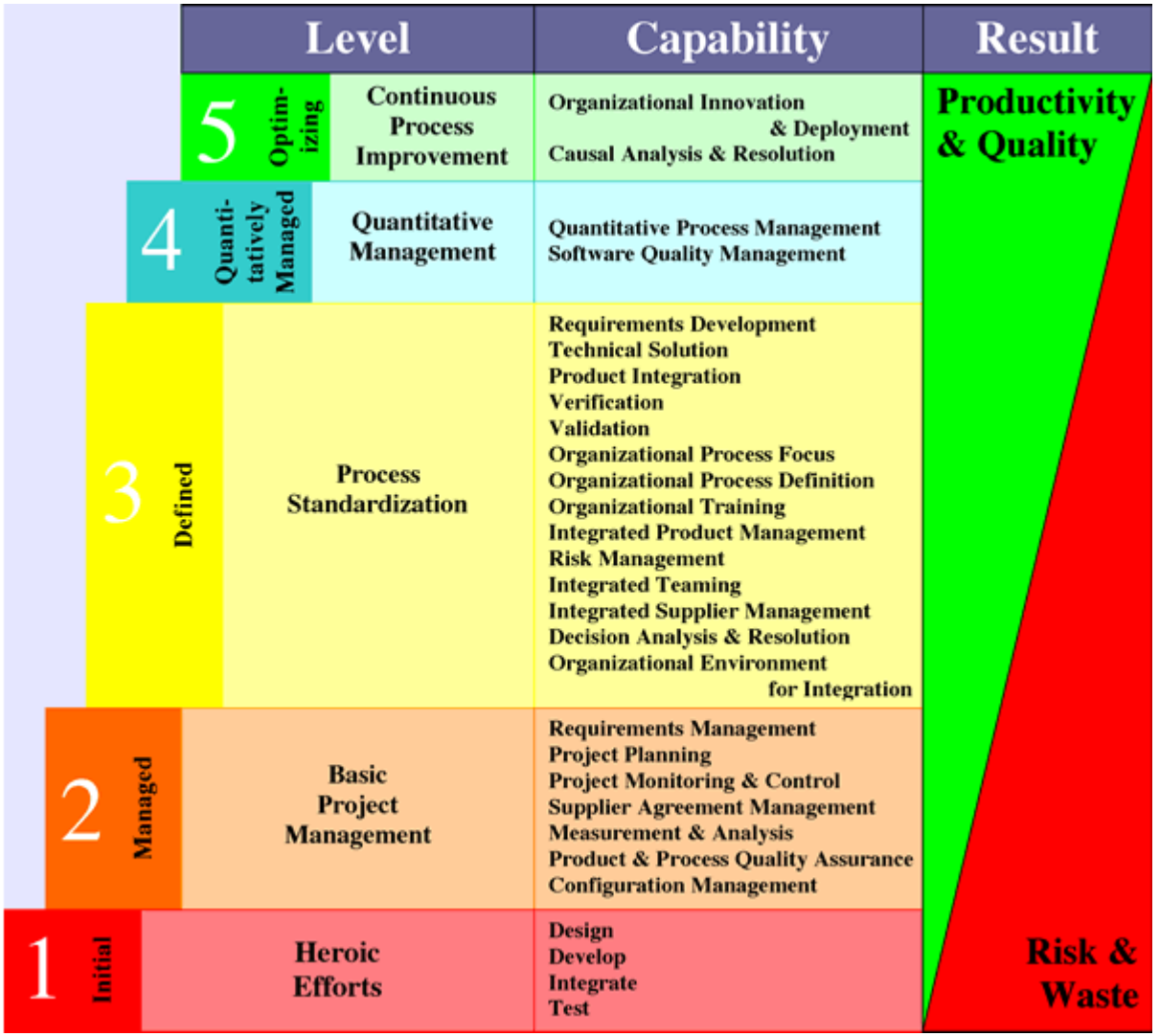


Rational Unified Process

- **A**dapt the process
- **B**alance stakeholder priorities
- **C**ollaborate across teams
- **D**emonstrate value iteratively
- **E**levate the level of abstraction
- **F**ocus continuously on quality

- Develop a Vision
- Manage to the Plan
- Identify and Mitigate Risks
- Assign and Track Issues
- Examine the Business Case
- Design a Component Architecture
- Incrementally Build and Test the Product
- Verify and Evaluate Results
- Manage and Control Changes
- Provide User Support

CMM / CMMI


Level
Capability
Result
5
Optim-izing
Continuous Process Improvement
Organizational Innovation & Deployment
Causal Analysis & Resolution
4
Quanti-tatively Managed
Quantitative Management
Quantitative Process Management
Software Quality Management
3
Defined
Process Standardization
Requirements Development
Technical Solution
Product Integration
Verification
Validation
Organizational Process Focus
Organizational Process Definition
Organizational Training
Integrated Product Management
Risk Management
Integrated Teaming
Integrated Supplier Management
Decision Analysis & Resolution
Organizational Environment for Integration
2
Managed
Basic Project Management
Requirements Management
Project Planning
Project Monitoring & Control
Supplier Agreement Management
Measurement & Analysis
Product & Process Quality Assurance
Configuration Management
1
Initial
Heroic Efforts
Design
Develop
Integrate
Test
Productivity & Quality
Risk & Waste

Agile Manifesto

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

· Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

· Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

· Business people and developers must work together daily throughout the project.

· Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

· The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

· Working software is the primary measure of progress.

· Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

· Continuous attention to technical excellence and good design enhances agility.

· Simplicity--the art of maximizing the amount of work not done--is essential.

· The best architectures, requirements, and designs emerge from self-organizing teams.

· At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile practices

- Scrum (sprints, product owners, daily scrumming, deliverables, retrospective)
- XP (pair programming ,refactoring, simplicity, courage)
- Crystal (people focus)
- DSDM (pareto principle, MoSCoW, good-enough delivery)
- Lean

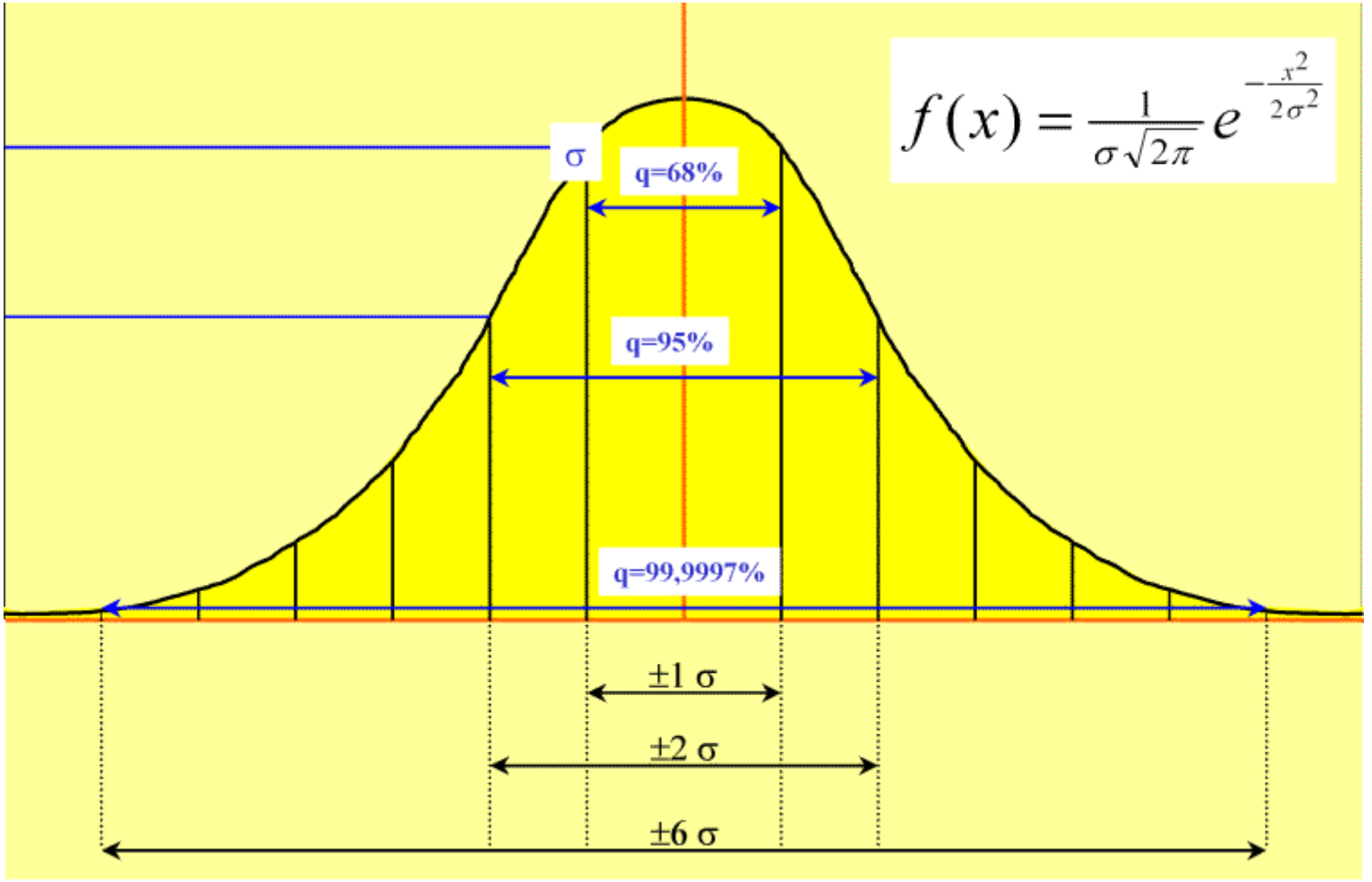
Lean software development

- Eliminate waste (redundancies, duplicate code, lack of clarity, etc) – identify through eg. code review, remedy through eg. generic programming / refactoring / mercilessness
- Amplify learning (small iterations, 1:1's, customer interaction)
- Decide as late as possible (small increments, RAD, thoughtful design, loosely coupled code)
- Deliver as fast as possible (clear goals, achievable goals, high velocity, daily scrumming, removing churn/context switching)
- Empower the team (trust, high-level instruction)
- Build integrity in (transparency, loyalty to the vision, clear focus)
- See the whole (system in context)

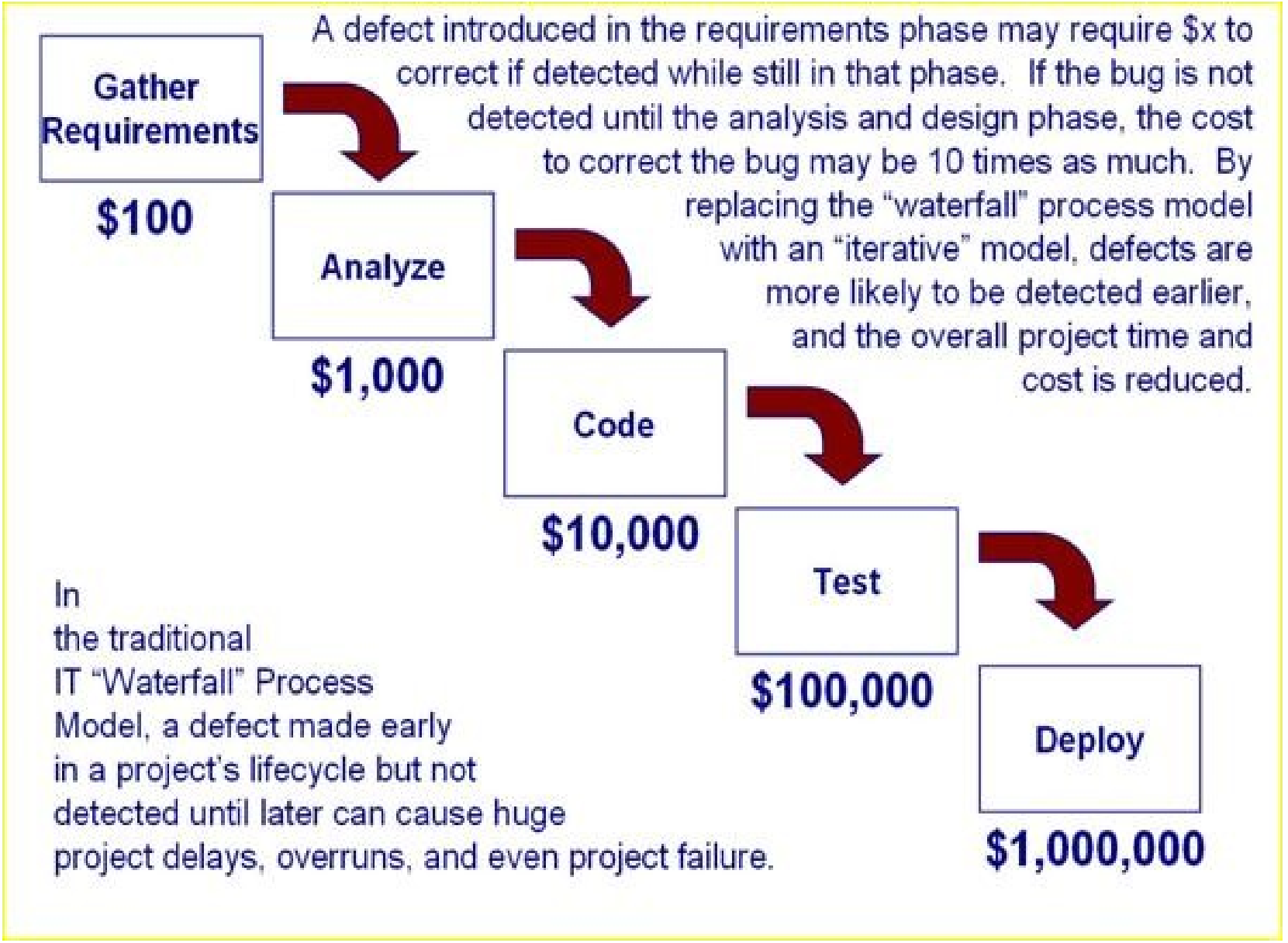
Deming's deadly diseases

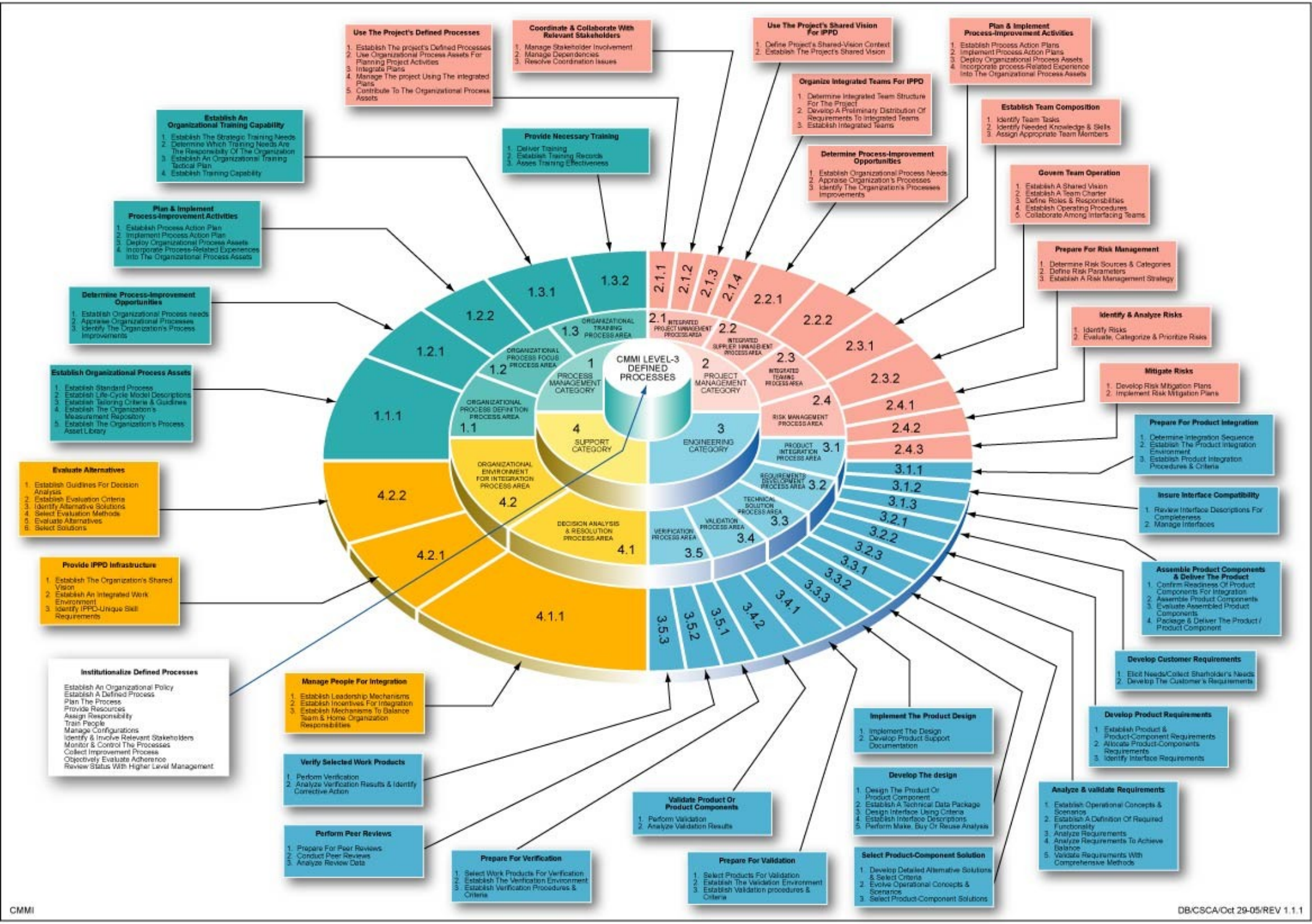
- Lack of constancy of purpose.
- Emphasis on short-term profits.
- Evaluation by performance, merit rating, or annual review of performance.
- Mobility of management.
- Running a company on visible figures alone.
- Excessive medical costs.
- Excessive costs of warranty, fueled by lawyers who work for contingency fees.

Six Sigma


$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$
σ
q=68%
q=95%
q=99,9997%
±1 σ
±2 σ
±6 σ

Cost of correction relative to phase




Use The Project's Defined Processes
1. Establish The project's Defined Processes
2. Use Organizational Process Assets For Planning Project Activities
3. Integrate Plans
4. Manage The project Using The integrated Plans
5. Contribute To The Organizational Process Assets
Coordinate & Collaborate With Relevant Stakeholders
1. Manage Stakeholder Involvement
2. Manage Dependencies
3. Resolve Coordination Issues
Use The Project's Shared Vision For IPPD
1. Define Project's Shared-Vision Context
2. Establish The Project's Shared Vision
Plan & Implement Process-Improvement Activities
1. Establish Process Action Plans
2. Implement Process Action Plans
3. Deploy Organizational Process Assets
4. Incorporate process-Related Experience Into The Organizational Process Assets
Organize Integrated Teams For IPPD
1. Determine Integrated Team Structure For The Project
2. Develop A Preliminary Distribution Of Requirements To Integrated Teams
3. Establish Integrated Teams
Establish Team Composition
1. Identify Team Tasks
2. Identify Needed Knowledge & Skills
3. Assign Appropriate Team Members
Establish An Organizational Training Capability
1. Establish The Strategic Training Needs
2. Determine Which Training Needs Are The Responsibility Of The Organization
3. Establish An Organizational Training Tactical Plan
4. Establish Training Capability
Provide Necessary Training
1. Deliver Training
2. Establish Training Records
3. Asses Training Effectiveness
Determine Process-Improvement Opportunities
1. Establish Organizational Process Needs
2. Appraise Organization's Processes
3. Identify The Organization's Processes Improvements
Govern Team Operation
1. Establish A Shared Vision
2. Establish A Team Charter
3. Define Roles & Responsibilities
4. Establish Operating Procedures
5. Collaborate Among Interfacing Teams
Plan & Implement Process-Improvement Activities
1. Establish Process Action Plan
2. Implement Process Action Plan
3. Deploy Organizational Process Assets
4. Incorporate Process-Related Experiences Into The Organizational Process Assets
Prepare For Risk Management
1. Determine Risk Sources & Categories
2. Define Risk Parameters
3. Establish A Risk Management Strategy
Determine Process-Improvement Opportunities
1. Establish Organizational Process needs
2. Appraise Organizational Processes
3. Identify The Organization's Process Improvements
Identify & Analyze Risks
1. Identify Risks
2. Evaluate, Categorize & Prioritize Risks
Establish Organizational Process Assets
1. Establish Standard Process
2. Establish Life-Cycle Model Descriptions
3. Establish Tailoring Criteria & Guidlines
4. Establish The Organization's Measurement Repository
5. Establish The Organization's Process Asset Library
Mitigate Risks
1. Develop Risk Mitigation Plans
2. Implement Risk Mitigation Plans
Prepare For Product Integration
1. Determine Integration Sequence
2. Establish The Product Integration Environment
3. Establish Product Integration Procedures & Criteria
Evaluate Alternatives
1. Establish Guidlines For Decision Analysis
2. Establish Evaluation Criteria
3. Identify Alternative Solutions
4. Select Evaluation Methods
5. Evaluate Alternatives
6. Select Solutions
Insure Interface Compatibility
1. Review Interface Descriptions For Completeness
2. Manage Interfaces
Provide IPPD Infrastructure
1. Establish The Organization's Shared Vision
2. Establish An Integrated Work Environment
3. Identify IPPD-Unique Skill Requirements
Assemble Product Components & Deliver The Product
1. Confirm Readiness Of Product Components For Integration
2. Assemble Product Components
3. Evaluate Assembled Product Components
4. Package & Deliver The Product / Product Component
Develop Customer Requirements
1. Elicit Needs/Collect Shareholder's Needs
2. Develop The Customer's Requirements
Develop Product Requirements
1. Establish Product & Product-Component Requirements
2. Allocate Product-Components Requirements
3. Identify Interface Requirements
Analyze & validate Requirements
1. Establish Operational Concepts & Scenarios
2. Establish A Definition Of Required Functionality
3. Analyze Requirements
4. Analyze Requirements To Achieve Balance
5. Validate Requirements With Comprehensive Methods
Select Product-Component Solution
1. Develop Detailed Alternative Solutions & Select Criteria
2. Evolve Operational Concepts & Scenarios
3. Select Product-Component Solutions
Develop The design
1. Design The Product Or Product Component
2. Establish A Technical Data Package
3. Design Interface Using Criteria
4. Establish Interface Descriptions
5. Perform Make, Buy Or Reuse Analysis
Implement The Product Design
1. Implement The Design
2. Develop Product Support Documentation
Validate Product Or Product Components
1. Perform Validation
2. Analyze Validation Results
Prepare For Validation
1. Select Products For Validation
2. Establish The Validation Environment
3. Establish Validation procedures & Criteria
Prepare For Verification
1. Select Work Products For Verification
2. Establish The Verification Environment
3. Establish Verification Procedures & Criteria
Perform Peer Reviews
1. Prepare For Peer Reviews
2. Conduct Peer Reviews
3. Analyze Review Data
Verify Selected Work Products
1. Perform Verification
2. Analyze Verification Results & Identify Corrective Action
Manage People For Integration
1. Establish Leadership Mechanisms
2. Establish Incentives For Integration
3. Establish Mechanisms To Balance Team & Home Organization Responsibilities
Institutionalize Defined Processes
Establish An Organizational Policy
Establish A Defined Process
Plan The Process
Provide Resources
Assign Responsibility
Train People
Manage Configurations
Identify & Involve Relevant Stakeholders
Monitor & Control The Processes
Collect Improvement Process
Objectively Evaluate Adherence
Review Status With Higher Level Management
CMMI LEVEL-3 DEFINED PROCESSES
1 PROCESS MANAGEMENT CATEGORY
2 PROJECT MANAGEMENT CATEGORY
3 ENGINEERING CATEGORY
4 SUPPORT CATEGORY
1.1 ORGANIZATIONAL PROCESS DEFINITION PROCESS AREA
1.2 ORGANIZATIONAL PROCESS FOCUS PROCESS AREA
1.3 ORGANIZATIONAL TRAINING PROCESS AREA
2.1 INTEGRATED PROJECT MANAGEMENT PROCESS AREA
2.2 INTEGRATED SUPPLIER MANAGEMENT PROCESS AREA
2.3 INTEGRATED TEAMING PROCESS AREA
2.4 RISK MANAGEMENT PROCESS AREA
3.1 PRODUCT INTEGRATION PROCESS AREA
3.2 REQUIREMENTS DEVELOPMENT PROCESS AREA
3.3 TECHNICAL SOLUTION PROCESS AREA
3.4 VALIDATION PROCESS AREA
3.5 VERIFICATION PROCESS AREA
4.1 DECISION ANALYSIS & RESOLUTION PROCESS AREA
4.2 ORGANIZATIONAL ENVIRONMENT FOR INTEGRATION PROCESS AREA
1.1.1
1.2.1
1.2.2
1.3.1
1.3.2
2.1.1
2.1.2
2.1.3
2.1.4
2.2.1
2.2.2
2.3.1
2.3.2
2.4.1
2.4.2
2.4.3
3.1.1
3.1.2
3.1.3
3.2.1
3.2.2
3.2.3
3.3.1
3.3.2
3.3.3
3.4.1
3.4.2
3.5.1
3.5.2
3.5.3
4.1.1
4.2.1
4.2.2