Software Process improvement (SPI)

- Discussion of topics to include:
  - The case for SPI
  - SPI Benefits and costs
  - Establishing a process programme
  - Process Focused Organizations
  - Process Discipline & Environment
  - Process Guides
  - Process Maturity and Frameworks
  - Process standards
  - Process measurement and assessment
  - SPI action planning
  - SPI standards

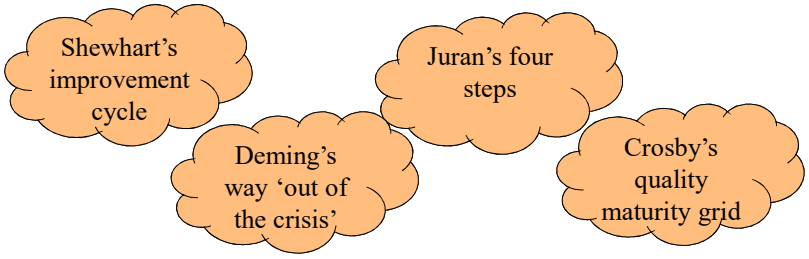
Software Process Maturity History

- In the 1970s and 80s we were concerned with process (lifecycle) models and structured methods
- From late 80s onwards, we are more concerned with software engineering as an engineering discipline
  - Become more process focused
  - Process improvement
  - Thus…
    - Process maturity
      - Process maturity frameworks

- 1930s
  - Walter Shewart
  - Statistical process control
- 1950s
  - Ed Deming
  - Statistical process control
- 1980
  - Phil Crosby
  - Quality maturity grid
- 1986
  - Watts Humphrey
  - Suggested levels of Process maturity
- 1990s
  - SEI, CMM, PSP, TSP
  - ISO, 9000, ISO 15504 (SPICE; now ISO/IEC 330XX)
- 2000…
  - CMMI, Agile
- 2010s…
  - Agile/lean/kanban???

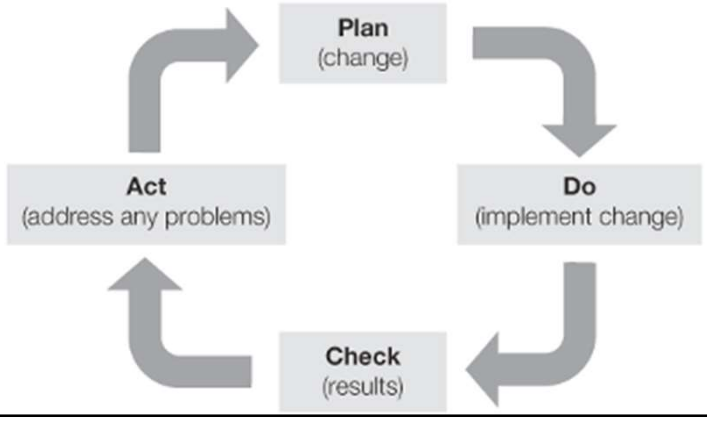
Learning from the Quality Gurus

- The Quality Gurus - Dr. W. Edwards **Deming**, Dr. Joseph **Juran**, Philip **Crosby**, Armand V. Feigenbaum, Dr. H. James Harrington, Dr. Kaoru Ishikawa, Dr. Walter A. **Shewhart**, Shigeo Shingo, Frederick Taylor, and Dr. Genichi Taguchi
  - Have made a significant impact on the world through their contributions to improving not only businesses, but all organizations including state and national governments, military organizations, educational institutions, healthcare organizations, and many other establishments and organizations.



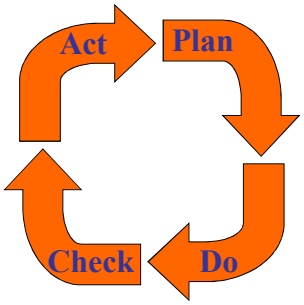
Shewhart cycle

- PDCA '**Plan-Do-Check-Act**' an iterative four-step management method to control and continuous improvement of processes and products.
- It is also known as the Deming circle/cycle/wheel, Shewhart cycle, control circle/cycle, or plan–do–study–act (PDSA).



Quality gurus
Shewhart's improvement cycle (1930s)

- Walter Shewhart proposed a '**Plan-Do-Check-Act**' cycle for quality improvement



1.Plan
- Define the problem
- State improvement objectives

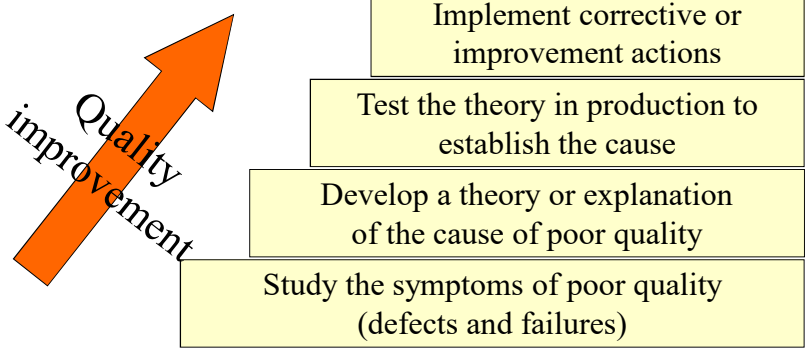
2.Do
- Identify possible problem causes
- Establish baselines
- Test changes

3.Check
- Collect data
- Evaluate data

4.Act
- Implement system change
- Determine effectiveness

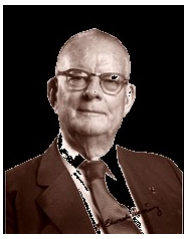
Quality gurus - Joseph Juran

- Juran's four steps:
  - Joseph Juran suggests a systematic approach to control and improve quality
  - Emphasizes the **management of quality at all phases** in the product lifecycle



Quality gurus – W. Edwards Deming

- Regarded by the Japanese as the chief architect of their industrial success
  - Deming's focus is on the **need for management's long-term commitment to new learning and new philosophy** in order to achieve transformation
  - **Quality is about people**, not products
  - "**All processes are vulnerable to loss of quality through variation**: if levels of variation are managed, they can be decreased and quality raised"
- Out of the Crisis (1984)
  - having a satisfied customer is not enough
  - profit in business comes from
    - repeat customers
    - customers that boast about your product and service
    - customers that bring friends with them
    - necessary to anticipate customer needs



Quality gurus – Philip Crosby

- 1984: Quality without Tears
  - "Do It Right First Time"
  - "Zero Defects"
  - Phil Crosby stresses the importance of involvement and motivation for everyone in the organization
- Crosby's quality maturity grid :
  - The Quality Management Maturity Grid (QMMG) is an organizational maturity matrix used by a business or organization as a benchmark of **how mature their processes are**, and **how well they are embedded in their culture**, with respect to service or product quality management.
  - QMMG states 5 maturity levels through which an organization or business will go through
    1. Uncertainty
    2. Awakening
    3. Enlightenment
    4. Wisdom
    5. Certainty

What have they taught us?

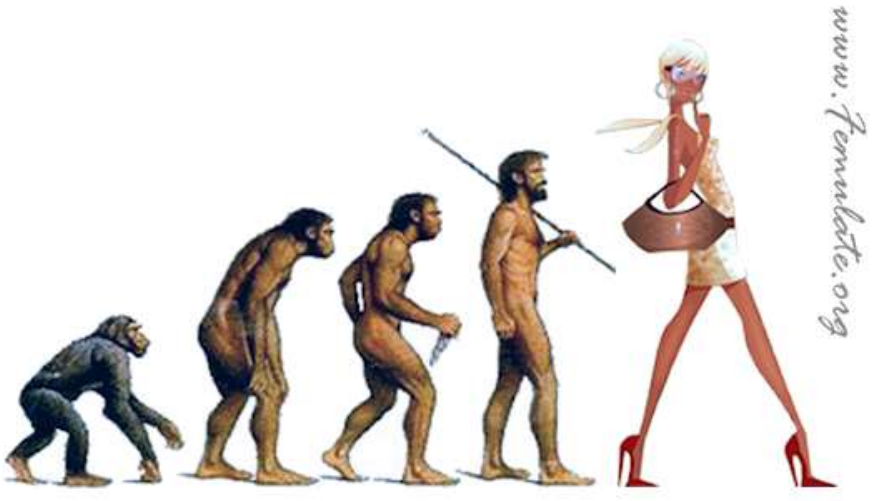
Software Process Improvement

- No matter which approach (waterfall, incremental, etc.) you take to developing software all the activities need to be managed within certain constraints
  - Thus **project management**
- We need a model for software development
  - Thus **process management**
- We need to improve the way we do it
  - Thus **process improvement**

Process change

- Involves making modifications to existing processes.
- This may involve:
  - Introducing new practices, methods or processes;
  - Changing the ordering of process activities;
  - Introducing or removing deliverables;
  - Introducing new roles or responsibilities.
- Change should be driven by **measurable goals**.

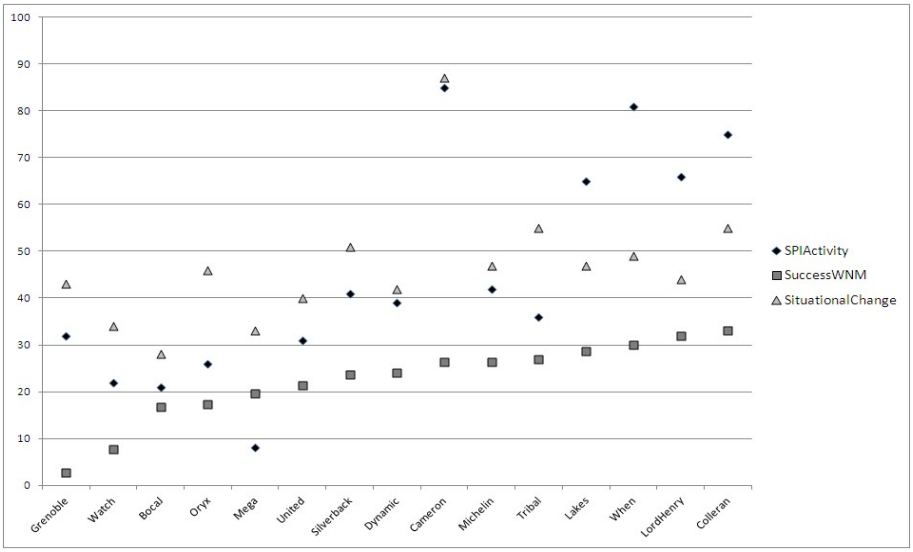
Change is inevitable...



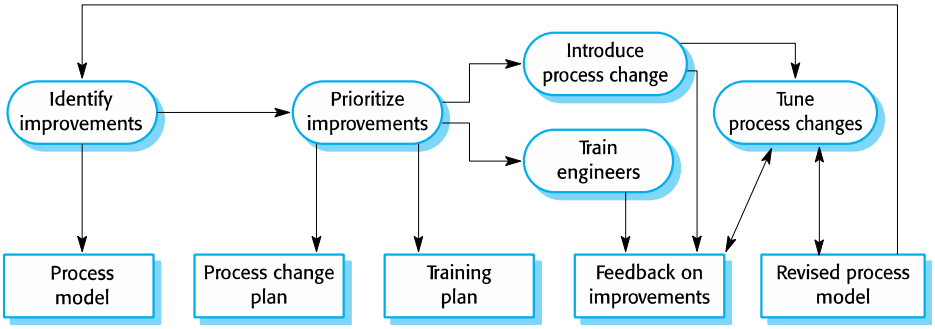
Evolutionary economics

- Proposes that *dynamic* or *adaptive* capabilities are important
- It is the ability to integrate new learning, to adapt in response to changing environments that will propel an organisation to optimised processes.
- Change is difficult?
- Is all change "good"?
- How often should process change take place?
- How much process change is appropriate?
- Should the process be changing constantly / regularly?

Software Process Adaptive Capability



The process change process



Process change stages

- Improvement identification
  - This stage is concerned with using the results of the process analysis to identify ways to tackle quality problems, schedule bottlenecks or cost inefficiencies that have been identified during process analysis.
- Improvement prioritization
  - When many possible changes have been identified, it is usually impossible to introduce them all at once, and you must decide which are the most important.
- Process change introduction
  - Process change introduction means putting new procedures, methods and tools into place and integrating them with other process activities.

Process change stages

- Process change training
  - Without training, it is not possible to gain the full benefits of process changes. The engineers involved need to understand the changes that have been proposed and how to perform the new and changed processes.
- Change tuning
  - Proposed process changes will never be completely effective as soon as they are introduced. You need a tuning phase where minor problems can be discovered, and modifications to the process can be proposed and introduced.

Process change problems

- **Resistance to change**
  - Team members or project managers may resist the introduction of process changes and propose reasons why changes will not work, or delay the introduction of changes.
  - They may, in some cases, deliberately obstruct process changes and interpret data to show the ineffectiveness of proposed process change.
- **Change persistence**
  - While it may be possible to introduce process changes initially, it is common for process innovations to be discarded after a short time and for the processes to revert to their previous state.

Resistance to change

- Project managers often resist process change because any innovation has **unknown risks** associated with it.
  - Project managers are judged according to whether or not their project produces software on time and to budget.
  - They may prefer an **inefficient but predictable process** to an improved process that has organizational benefits, but which has short-term risks associated with it.
- Engineers may resist the introduction of new processes for similar reasons, or because they see these processes as **threatening** their professionalism.
  - That is, they may feel that the new pre-defined process gives them less discretion and does not recognize the value of their skills and experience.

Change persistence

- The problem of **changes** being introduced then subsequently **discarded** is a common one.
  - Changes may be proposed by an 'evangelist' who believes strongly that the changes will lead to improvement. He or she may work hard to ensure the changes are effective and the new process is accepted.
  - If the 'evangelist' leaves, then the people involved may therefore simply revert to the previous ways of doing things.
- Change **institutionalization** is important
  - This means that process change is not dependent on individuals but that the changes become part of standard practice in the company, with company-wide support and training.

So Now What?

- How do we manage change and improvement?
  - We need a Roadmap / Framework of some kind to implement SPI
- What else do I need to know?
  - Key concepts of Process Maturity and Capability
  - What frameworks exist and how they work…
  - How to choose one?

Key Underlying Concepts

- **Process Capability** refers to the ability of a process to achieve a defined goal
- **Process Maturity** refers to the extent to which a specific process is explicitly defined, managed, measured, controlled, and effective
- **Organisational Maturity** refers to the overall capability of the organisation, across all software development processes.

Philosophical Background

- The concepts of **Capability** and **Maturity** are important because software development projects must adhere to:
  - Deadlines
  - Costs / Budgets
  - Quality Level
  - Functional/Content commitments
- In order to satisfy these constraints, the process must be:
  - **Predictable** (we can predict the duration, cost, content and quality of a product)
  - **Repeatable** (we can be predictable over a period of time and on many projects).
- The more capable a software development process is, the more predictable it is...

Process maturity

- Organisations at various levels of process discipline are like people at different levels of maturity
- **Software process maturity is the extent to which a specific software process is explicitly defined, implemented, managed, measured, controlled, and effective.**
  - Maturity implies a potential for growth in capability and indicates both the richness of an organization's software process and the consistency with which it is applied in projects throughout the organization.
  - The process is well-understood throughout a mature organization, usually through documentation and training, and the process is continually being monitored and improved by its users.
- **The capability of a mature process is known.**
  - Software process maturity implies that the productivity and quality resulting from an organization's software process can be improved over time through consistent gains in the discipline achieved by using its software process.

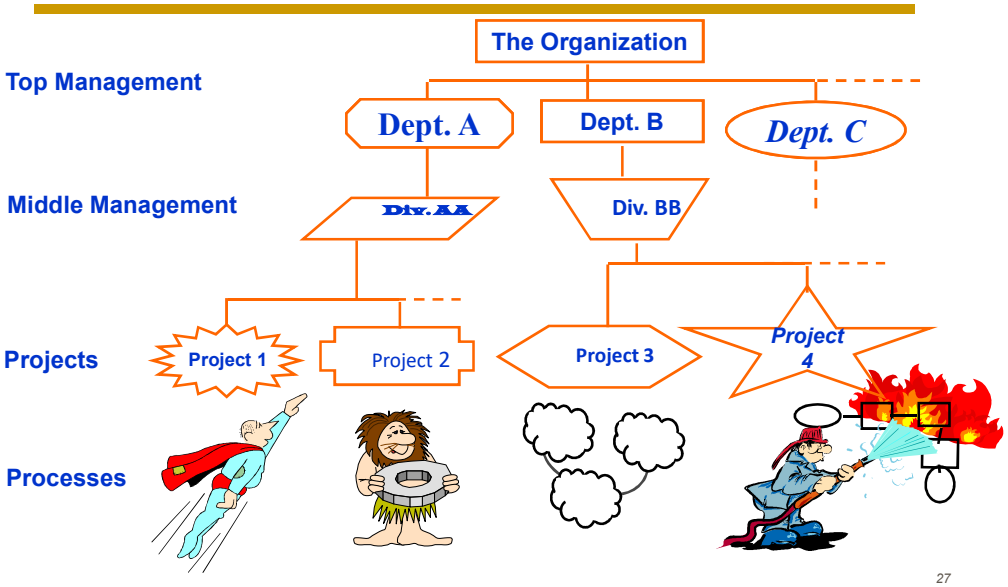
Immature Organizations

- Processes are ad hoc, and occasionally chaotic.
- Processes improvised by practitioners.
- Testing and reviews usually curtailed under stress.
- Quality is unpredictable.
- Costs and schedules are usually exceeded.
- Reactionary management is usually firefighting.
- Success rides on individual talent and heroic effort.
- Technology benefits are lost in the noise.
- BUT this does not necessarily mean that immature is inherently 'bad' – it depends!

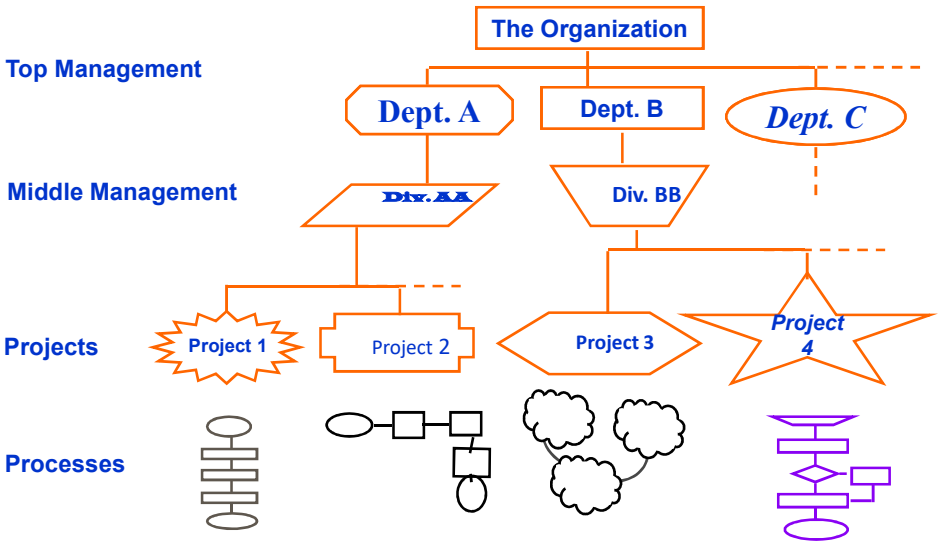
Mature Organizations

- Processes are **defined** and **documented**.
- Management plans, **monitors**, and communicates.
- **Roles** and **responsibilities** are **clear**.
- **Product** and **process** are **measured**.
- Quality, costs, and schedules are **predictable**
- Management committed to continuous improvement.
- Technology used effectively within defined process.
- BUT this does not necessarily mean that mature is inherently 'good' – it depends!

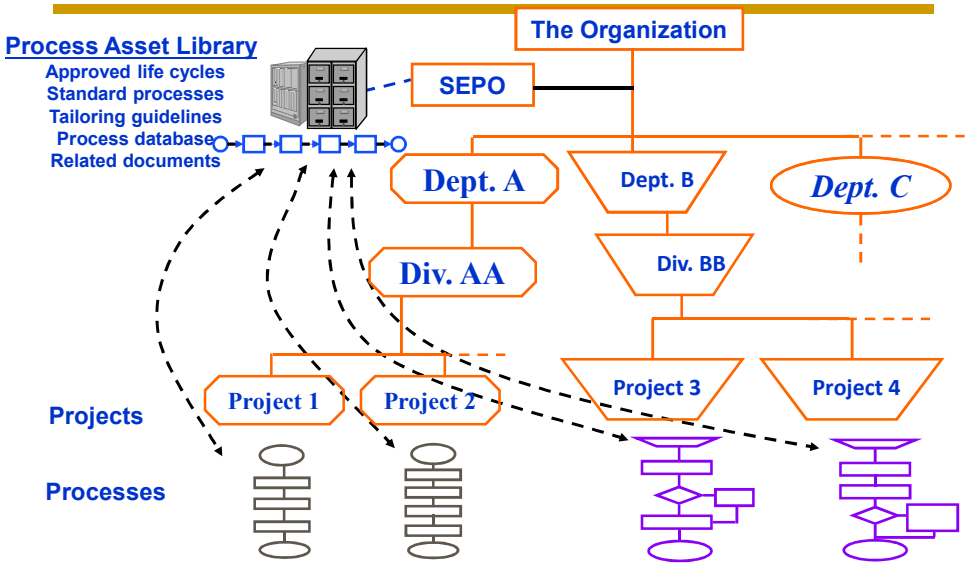
Organization with few processes in place



Many processes in place… but they are project-specific



Sample Higher-Level Organization processes based on organization's Process Asset Library (PAL)



Software Process Improvement Models

- The influence of Shewhart, Juran and Crosby can be traced easily in the work published on SPI
- Watts Humphrey pioneered SPI and contributed greatly to the concept of software process maturity
  - He laid the foundation for the creation of the CMM (and CMMI)
    - But more on that later
- In early work Humphrey suggested 6 steps for organisation to improve their software process capabilities:
  1. Understand current status of their processes
  2. Develop a vision of the desired process
  3. Establish a list of required process improvement actions in priority order
  4. Produce a plan to accomplish these activities
  5. Commit resources to execute the plan
  6. Restart at step 1

Characteristics of modern SPI Models

- Most if not all modern SPI models and standards are directly related to Humphrey's 6 steps
- Many Standards / Roadmaps / Frameworks to choose from
  - Trillium; Bootstrap; CMMI; ISO 9000; ISO/IEC 15504; ISO/IEC 330XX, ISO/IEC 29110 and more…
- All have common characteristics…

Framework for SPI

- The main components of any SPI framework are:
- **Software process infrastructure**
  - Organisation and management infrastructure
  - Technical infrastructure
- **Software process improvement roadmap**
  - Specify a model characterising the software process
  - Could be one of the public roadmaps – 330XX or CMMI
- **Software process assessment method**
  - Methods and techniques for assessing current process & practice
  - Performed against a roadmap – 330XX or CMMI
- **Software process improvement plan**
  - Transform assessment findings into SPI actions

Retrospectives Versus SPI

- SPI frameworks are not commonly used across the broader framework of software development.
- The infrastructure, assessments, and management approach are more in the mode of traditional software development.
- In markets where predictability and reliability are demanded, the formalised SPI frameworks are heavily employed.

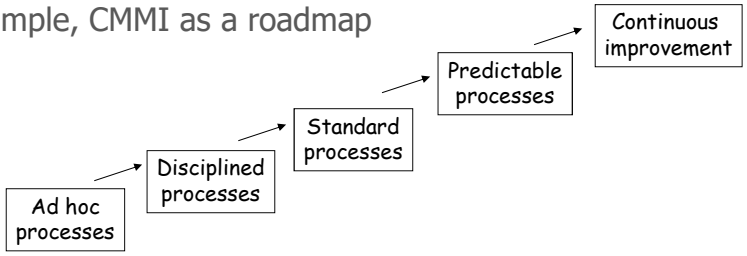
- But how does SPI occur in settings where SPI frameworks are not employed?
  - Retrospectives
  - Regular communications
  - Planning meetings
  - Side effect of various agile practices, e.g. code reviews could identify targets for SPI.

Software process infrastructure

- To establish an effective infrastructure, a software process environment must have 2 types of infrastructure:
- **1. Organisation and management infrastructure**
  - Roles and responsibilities for establishing, monitoring and enforcing the process activities
  - Typical roles:
    - Sponsor – senior executive support
    - Management – steering committee for strategic guidance
    - Coordination (SEPG) – coordinator and technical guidance to PITs
    - Teams – process improvement teams (PIT) perform the SPI activities
- **2. Technical infrastructure**
  - Technical platforms, facilities and tools that support the Software Engineering Process Group (SEPG) and Process Improvement Teams (PITs)

Software process improvement roadmap

- Roadmap defines **successive levels of process maturity**
- A roadmap should:
  - Have explicit steps to demonstrate achievement of process maturity
  - Define and list process attributes for each level
  - Define attainment attributes for each level
  - Identify challenges at each level
- For example, CMMI as a roadmap



Software process assessment method

- Assessment provides **a baseline of current status**
  - Should uncover the real state of software process in the organisation
- An organisation needs a baseline against which to measure improvement
- Assessment methods are normally based on process model & roadmap and **assess the organisation against the model and improvement scale**
- Normally conducted by **trained team of software practitioners**.
  - In Ireland, for example see www.cse.dcu.ie
- Areas covered by assessment include:
  - Process documentation, process training, inspection of process results, feedback, improvement mechanisms, roles and responsibilities

Software process improvement plan

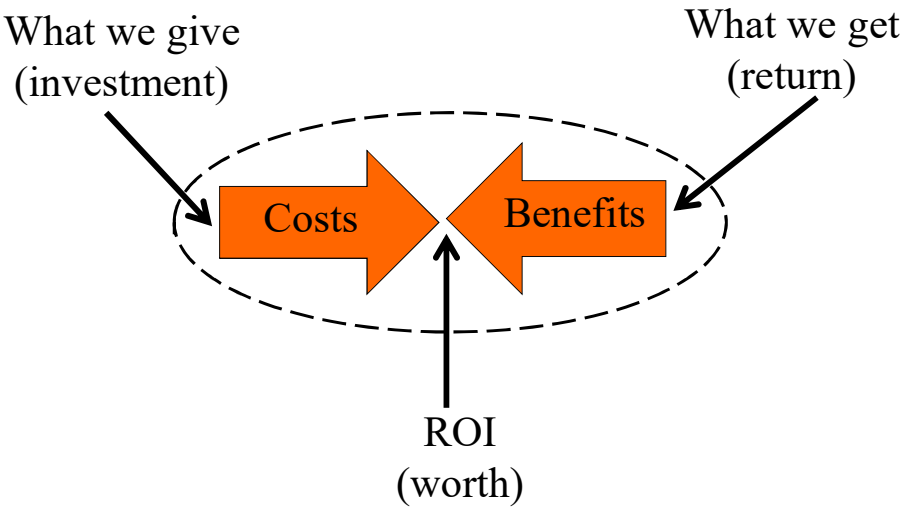
- SPI plan should be **based on assessment results** and should be treated as **a project in its own right**
  - ie. Given staff, budget, deadlines, etc.
- It must be **sponsored by management**
- Change should be viewed in the context of organisation and current organisational **context**, objectives, strategic plan, etc.
- It must **involve** everyone in the software organisation
- SPI is **continuous**

Process Maturity and Standards

- CMMI - Capability Maturity Model Integration
  - Formally CMM (Capability Maturity Model)
  - Can be used in three different areas of interest:
    - *CMMI for Development (CMMI-DEV)*
    - CMMI for Acquisition
    - CMMI for Services
- ISO offerings…
  - ISO 9000
  - ISO/IEC 15504 (SPICE)
  - ISO/IEC 330XX (SPICE mark II)
  - ISO 29110 (VSE)

Do they [SPI models] work?

Return on investment



Benefits of SPI

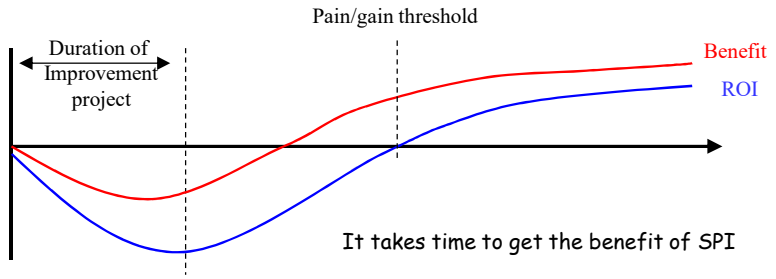
- One of the major obstacles for the adoption of SPI is the reluctance of business management to invest in SPI because they **do not have convincing evidence of the return on investment**
- There is a general **lack of reliable information** on the business benefits of SPI and there is little hard evidence of satisfactory ROI from SPI
- Often left unmeasured and therefore hard evidence is limited
- Questions we must ask are
  - Can we measure improvements in our software processes?
  - Can we measure subsequent benefits to our business?
  - Can we show a causal relationship?

What are the benefits?

- First major benefit from SPI is it **makes you think** about what you are doing and why you are doing it
  - The days go on and we follow the same basic habits
  - A commonly used practice was started by someone who is long gone and so is the rationale for the practice
- Typical benefits:
  - Decreased reliance on testing
  - Improved teamwork
  - Reduced rework
  - Efficient project staff start-up time
  - Reduced development costs
  - Improved predictability
  - Improved tool usage
  - Faster project start-up

Documenting results of SPI

- There are many difficulties with collecting data on SPI:
  - Many companies **don't keep the beginning data**. So when they improve, what are they comparing the new figures to?
  - There is **no standard way to measure costs or benefits**
  - **It takes time to see the full story**… you have to keep working at it long enough to really see the benefits



Justifying company investment in SPI

- It is generally **difficult to justify a ROI (Return-On-Investment) for SPI in monetary terms**.
  - Sound management and engineering practices should not have to be ROI issues.
  - Organisations with no process discipline don't have enough data to develop ROI arguments
  - ROI arguments ignore intangible benefits
- Process improvement programs are shown in a series of reports to reduce development costs and rework costs, as well as to improve productivity, cycle time and quality
- These improvements, once implemented by an organization, are shown to have a significant positive ROI to the improved organization

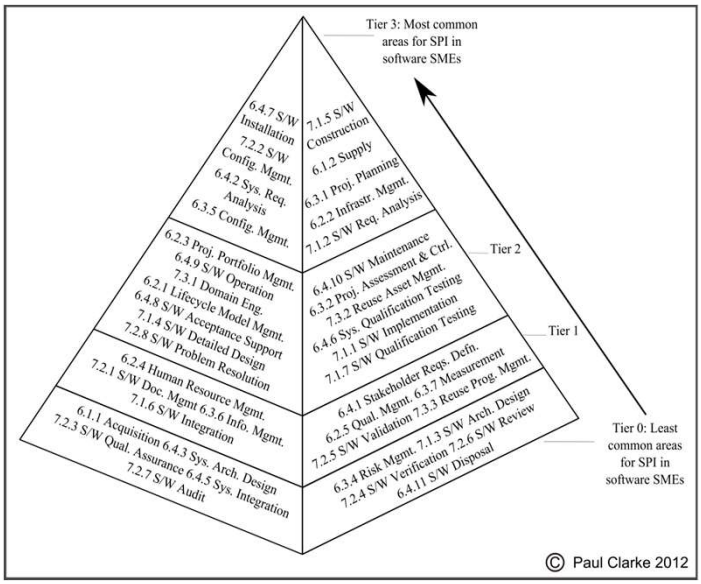
Costs / Benefits of SPI

- **Typical Costs**
  - Training fees, labor hours, travel costs
  - Policies, procedures, processes, life cycles
  - Project costs, activity costs, administration
  - Documents, reports, records, memos
  - Indoctrination costs of custom processes
  - Response conditioning costs
  - Mock appraisal costs, appraisal costs
- **Typical Benefits**
  - Higher quality
    - fewer defects
  - Lower maintenance
    - less rework
  - Higher productivity
    - low development cost
  - Faster cycle times
    - quick time-to-market
  - Greater value
    - more product features
  - Greater variety
    - more product variations
  - High customer satisfaction
    - more contracts

Common Myths of SPI ROI

- Software process improvement has no ROI
- Process improvement takes a long time
- Process improvement is too expensive
- Process improvement can't be performed in a few hours, days, weeks, or months
- Process performance can't be measured in only a few hours, days, weeks, or months
- Process improvement is only for large, mission critical programs

The nature of SPI in smaller companies

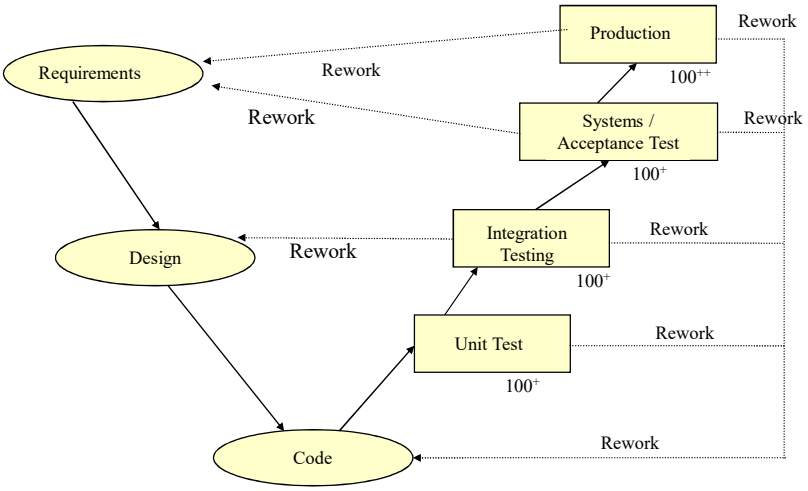


Example

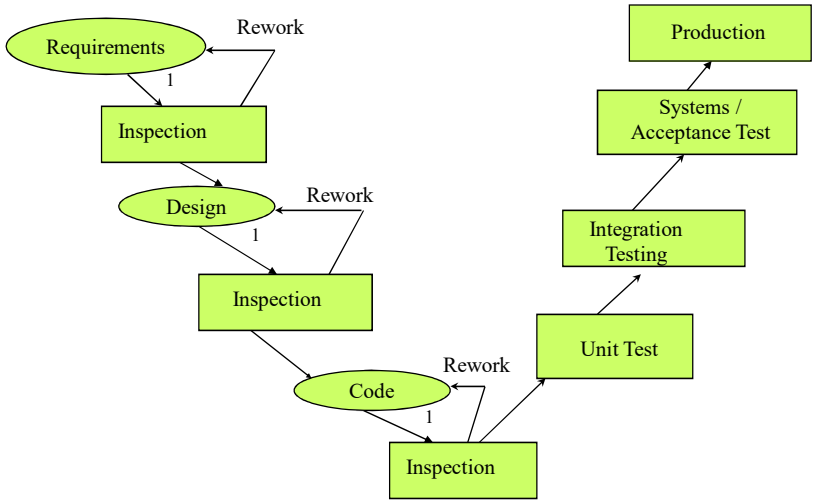
- Think about a typical Software Process with and without inspections

- What is the effect?
- How much does it cost?
- Is the extra process discipline worth it?

SPI Example 1: Software Process without Inspections



SPI Example 1: Software Process with Inspections

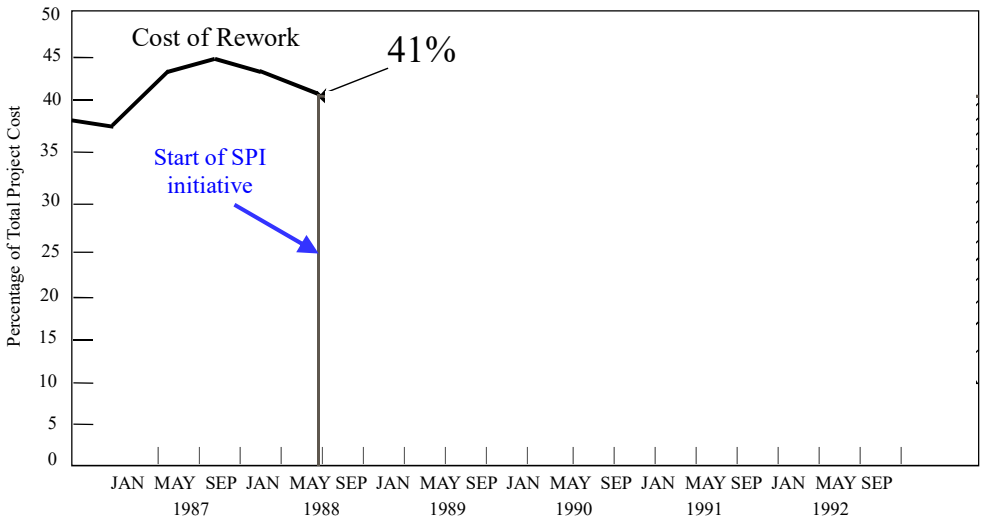


Case Study example: Raytheon

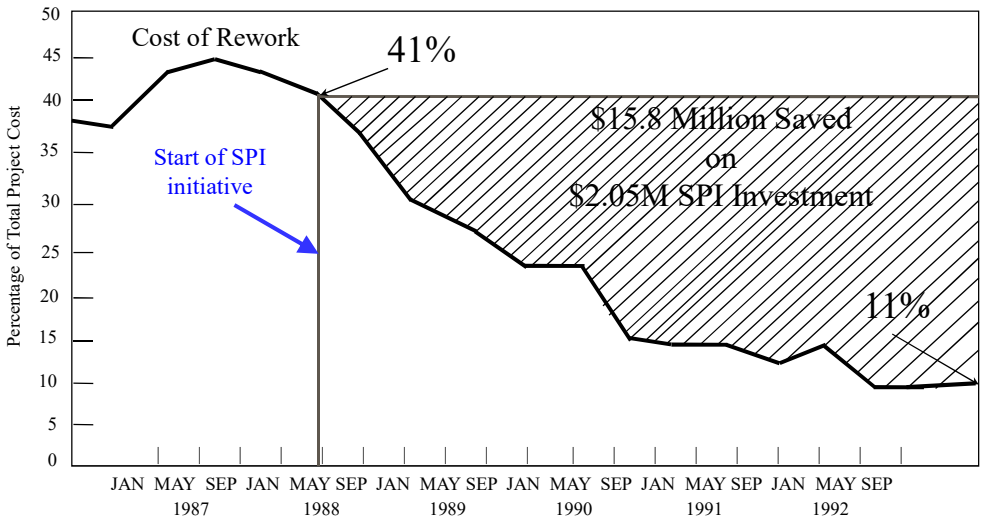
- A "famous" case study… old but used as a reference example
- Business Need: Raytheon concluded "their processes weren't adequate to deal with the diverse needs of customers."
- Proposed Solution: CMM Level 3

Source: Robert .B. Grady, Successful Software Process Improvement, Prentice-Hall, 1997.

Case Study: Raytheon Savings due to CMM level 3 SPI Efforts!



Case Study: Raytheon Savings due to CMM level 3 SPI Efforts!



Case Study: Raytheon (Cont..)

- Raytheon SPI Results show:
  - 7.7 to 1 ROI; achieved SEI/CMM Level 3.
  - Two-fold increase in productivity.
  - Cost of rework reduced from 41% to 11% of total project cost.
  - Eliminated $15.8M in rework due to reducing error-rate at a $1,281 SPI cost per engineer per year.

Not all SPI is successful

- SPI projects suffer greatly if they do not have senior management sponsorship
- SPI projects also suffer greatly if they do not have the support of those affected by the process change.
  Steps to avoid this
- Budget allocated to SPI projects
- Results of the project are measured and communicated to stakeholders