UNIVERSITY OF LEICESTER

Software Processes and Quality

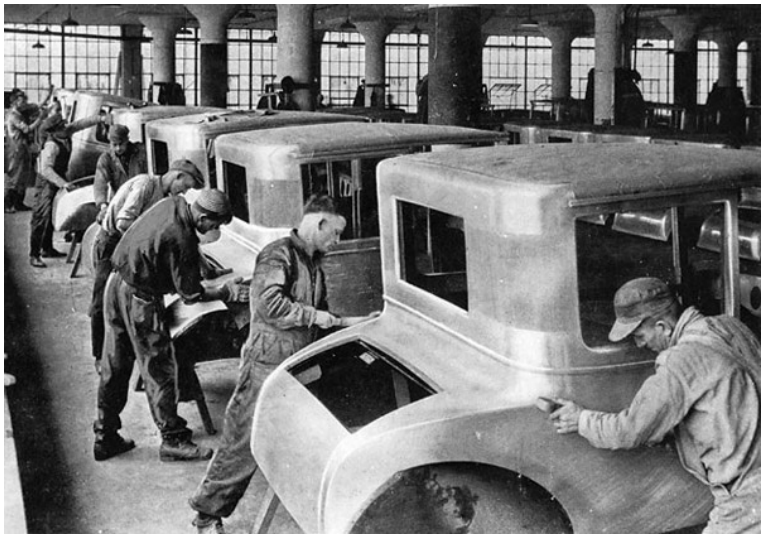
CO3095, CO7095, CO7508

José Miguel Rojas – *material provided by Neil Walkinshaw*

Roots in Manufacturing

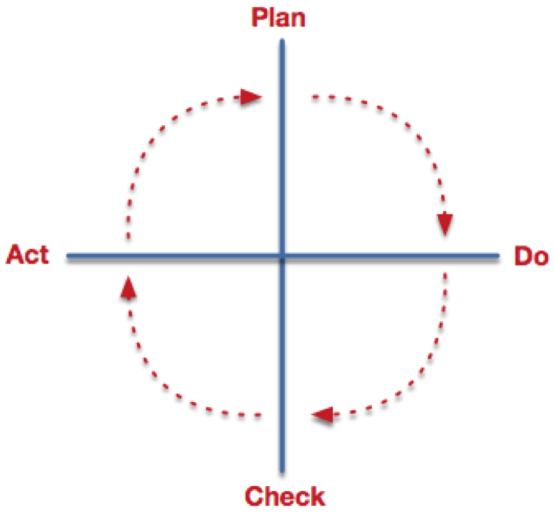
Assembly Lines

- Late 19th, early 20th Century
- Breaking a process into simple steps
- Easier to train individuals to repeat simple steps
- Enabled production to scale up.
- Easy to create multiple factories with simple rules.
- **Direct link between processes and quality / profitability.**



Plan - Do - Check - Act (1931)

- Walter Shewhart worked for Bell Telephones, addressing variabilities in their phone line quality
- Devised a technique for “Process Improvement”.
  - Plan: Set goals for project
  - Do: Carry out planned procedures
  - Check: Review problems
  - Act: Identify remedies for problems.
- Key goal - **reduce variability** of quality.



Post-War Japan as an Economic Power

- After the war, Japan rebuilt itself as an economic power.
- Recruited many experts in quality assurance
  - Including Shewhart and Juran, amongst others.
  - Fed into the Japanese “economic miracle”.
- Characterised by the Toyota Production System (TPS)
  - Revolved around the mantra of “eliminating waste”.
  - Active part of Toyota production to this day.

Toyota Production System

Plan	**Heijunka** - have the correct number of parts required to build a vehicle **Nemawashi** - decisions should be arrived at as a team, not dictated by individuals **Genba** - understand work load on individuals, ensure processes are transparent
Do	**Andon** - enable workers to immediately highlight threats via alert mechanisms **Just-In-Time** - Products should only be developed if and when they are required **Jidoka** - Capture faults as close to their source as possible **Genchi Genbutsu** - the best way to solve a problem is to see it for yourself **Kanban** - a board to communicate the state of the manufacturing system
Check	**Kaizen** - Staff encouraged to look for areas of improvement. **Muda, Muri, Mura** - eliminate waste.
Act	**Poke Yoke** - Prevent errors from occurring by embedding prevention mechanisms. **Hansen** - Learn from mistakes to prevent them from recurring

Maximising opportunity for feedback.

UNIVERSITY OF LEICESTER

Crisis of Confidence in US & European Manufacturing

- Japanese prominence accompanied by European & US stagnation
- UK became a net importer of goods for the first time
- Significant failures, such as the Challenger disaster (1986)
- Significant, widespread introspection, led to new quality assurance initiatives

Challenger launch was widely publicised, exploded on launch with six astronauts aboard, including a school teacher. Found to be the result of multiple quality-assurance failures

Total Quality Management (TQM)

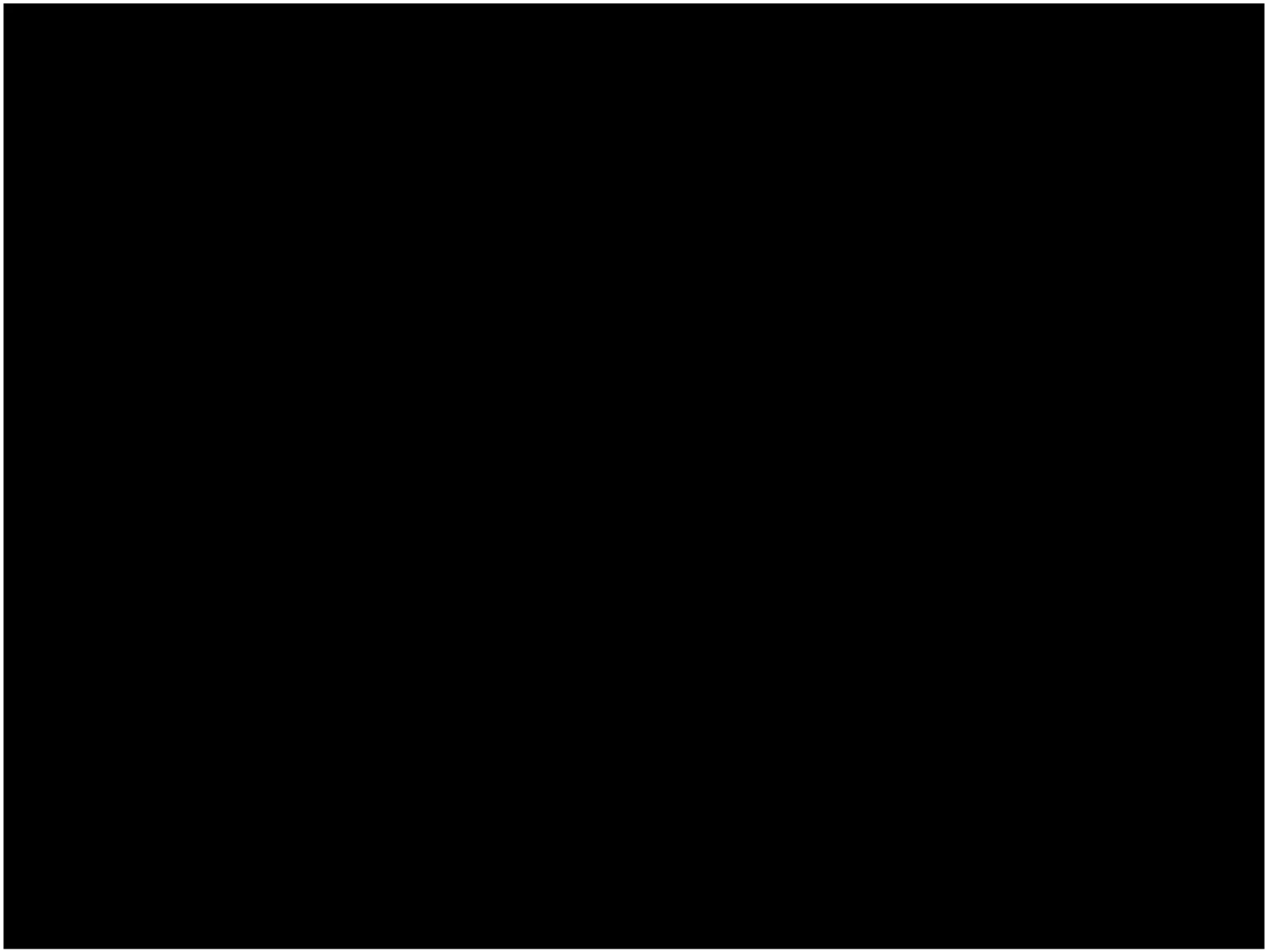
- Broad family of techniques, developed in the UK & US
- Follow the following stated principles:
  - Customer focus: Main objective is to achieve **total customer satisfaction**.
  - Process: Reduce process variations, achieve **continuous improvement**.
  - Human side of quality: Create a **company-wide culture**, from leadership down to employee empowerment.
  - Measurement and analysis: Drive improvement by **continuous measurement**.
- Inspired several derivative process improvement techniques
  - ISO9000, Motorola's Six Sigma, …

How is this related Software?

- The roots of [software] process improvement lie in the manufacturing industry
- Initiated by Shewhart's PDCA cycle
- Contributed to Japanese economic boom
  - Toyota Production System
- Spurred modern quality assurance frameworks
  - TQM, ISO9000, Six Sigma
- Formed the basis for typical software process improvement

The Traditional View of “Programming”

Digital Computer Techniques: Programming (AT&T, 1962)



As Simple as “A,B,C”

- Just draw a flow-chart and code it in.
- Programming boils down to thinking about “the logic”.

How does this relate to the quality models we discussed last time?

Notable Successes

- Despite this reductive view of software development.
- Apollo 11 control software
  - First manned moon landing (1969)
- Developed by an MIT team led by Margaret Hamilton
- ~450,000 LOC
- Included advanced concurrent algorithms

Software Crisis

- Coined at the 1968 NATO Conference on Software Engineering

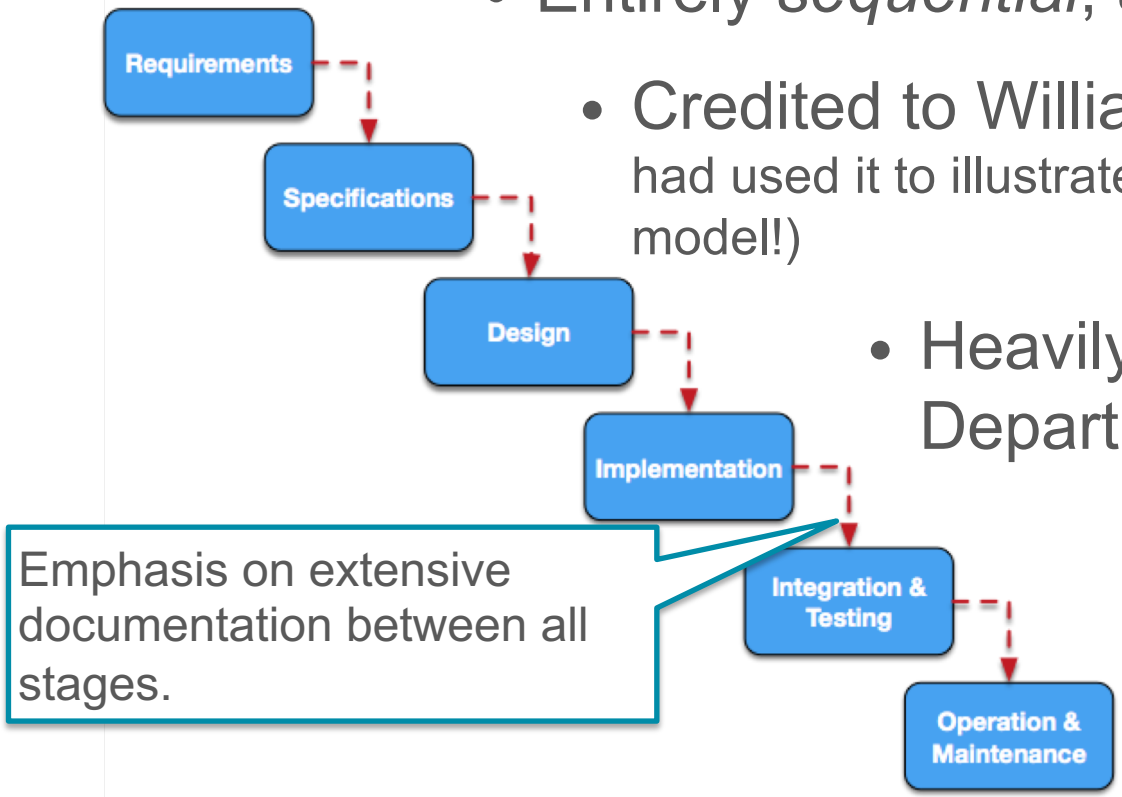
"The major cause of the Software Crisis is that machines have become several orders of magnitude more powerful! To put it bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming was a mild problem and now we have gigantic computers, programming has become an equally gigantic problem."

Edsger Dijkstra, 1972 Turing Award lecture

Development Processes

The Waterfall Model (1970)

- First popular effort to structure software development
  - Entirely *sequential*, *staged* process
    - Credited to William Royce (even though he had used it to illustrate an unsuitable development model!)
    - Heavily promoted by the US Department of defence



Problems with Waterfall

- What if requirements changed?
  - Could only be rectified at huge cost
- Stakeholders were only involved at the start
  - Could not affect change later on
- Inflexible partitioning of project in rigid, sequential phases

Iterative & Incremental Software Development (IID)

Iterative

- Software development takes place in iterations.
  - Each iteration is a refinement of the previous version.
- Each iteration is time-boxed.

- **Rapid feedback to / from stakeholders.**
- **Easy to incorporate change.**

Incremental

- Do not try to implement all of the features in the product at once.
  - One feature at a time.

- **Focus on prioritising key functionalities.**
- **Ensure a functioning product throughout.**
- **Succeed by a process of "divide and conquer".**

How does this relate to Joseph Juran and Phil Crosby's views of quality?

The Roots of IID

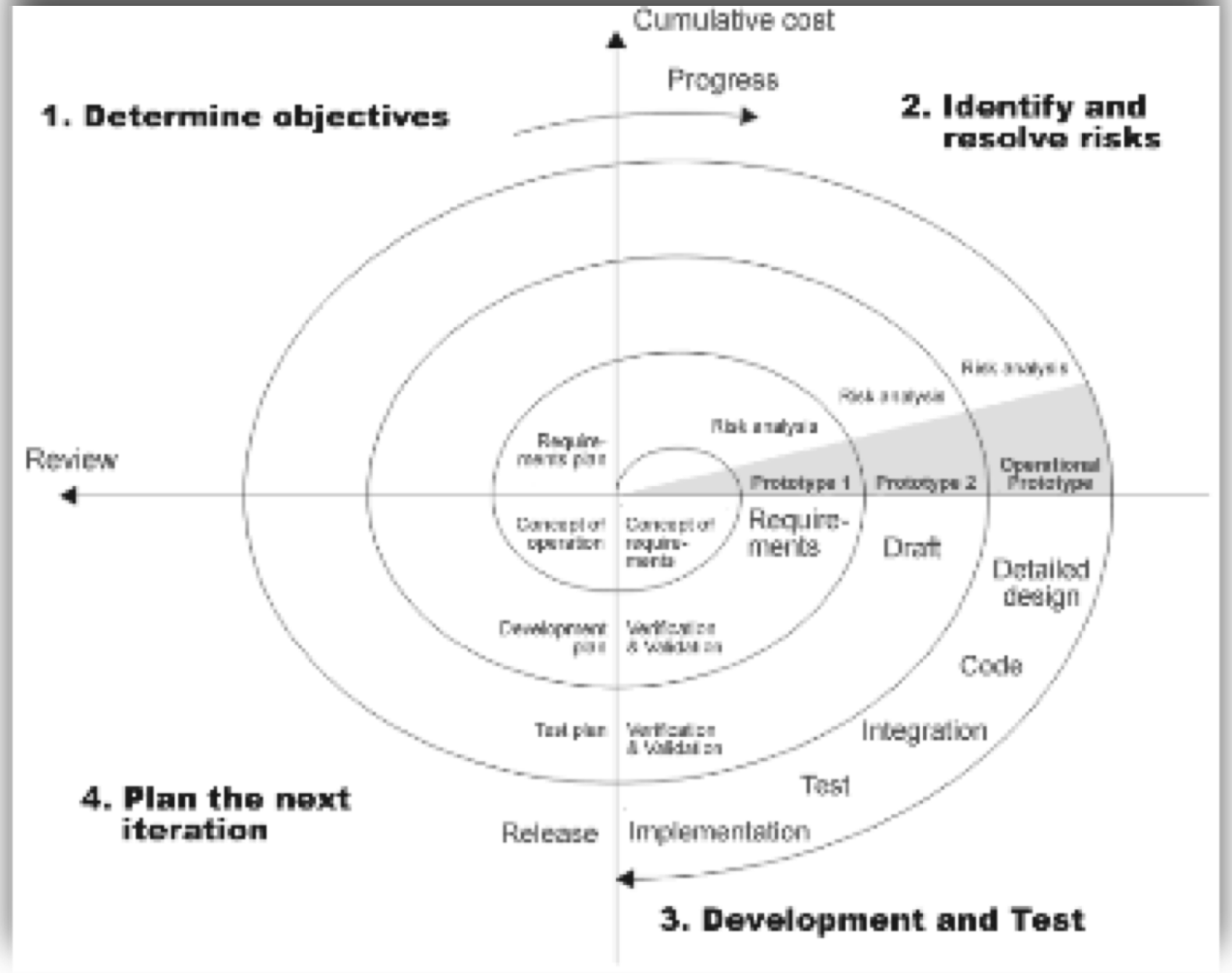
- Developed at NASA in the 1950s.
- Major success with work on their experimental X-15 jet.
- Became widely adopted within IBM and US Department of Defence.
- Time boxing was valuable.
  - Development of C&C systems on Trident Nuclear submarines.
  - Fine of $100,000 per day overdue.

Still holds the record for the highest speed recorded by an aircraft with a pilot. (Mach 6.72; 7,200 km/h).

Gradual Dominance of IID

- Waterfall model remained dominant throughout the 80s.
- Gradually supplanted by IID-inspired development models
  - Boehm's Spiral Model
  - Rational Unified Process (developed within IBM)
  - Agile software development (more on this later).

Spiral Model



Processes are merely “frameworks”

No “One Size Fits All” Solution

Summary

- Until the 70s, software development was perceived as merely “programming” (simple as A,B,C)
- Waterfall model was widely adopted
  - Shown to have many weaknesses
- Gradually replaced by IID
  - More versatile, easier to control
  - Formed basis for today’s popular development methodologies
- These can only form broad guidelines
  - Exact process needs to be tailored to suit development context.