Overview

Explain what Software Engineering (PPM) is all about

Module Specification

- The difference between writing a **small program** and developing a **large-scale software product** is not a difference of **size** as much as it is a difference of **nature**.
- While the construction of small programs is a purely technical issue, the development of larger software products is a multi-dimensional problem, which involves **technical** considerations as well as **managerial** and **organizational** considerations.
- **The objective of this module is to explore all three inter-related perspectives and to provide the learner with the foundational skills necessary to understand and apply these skills in an industry context.**

Transitioning from Uni to Workplace (1)

Vs

Typical Student Assignments:

- Rudimentary examination of aspects of quality.
- Absence of focus on cost.
- Static requirements for the most part.
- Small scale (teams and LOC).
- Limited communication issues.
- No real world paying users.

Commercial S/W Dev:

- Robust examination of quality.
- Explicit focus on cost.
- Volatile requirements.
- Larger scale (teams and LOC).
- Communication challenges.
- Paying users.

Transitioning from Uni to Workplace (2)

Vs

Typical Student Assignments:

- No customer management.
- Limited operational/deployment concerns.
- Rarely adapted/maintained once completed.
- Mostly steady-state teams / individuals.
- Short term, technology static.
- Impact of poor quality software is limited: low grades.

Commercial S/W Dev:

- Active customer management.
- Live service/product (SLAs)
- Mostly maintained/adapted into the future.
- Staff turnover
- Medium-long term, technology changing.
- Potentially very high impact: reduced salary/bonus, no job, financial penalties, incarceration (in extreme cases)

Transitioning from Uni to Workplace (3)

Commercial S/W Development:

- The **testing and quality assurance** aspects of the process must be adequate to avoid costly instances of absence of quality in operational environments.
- **Information on efficiency and productivity** of work are provided by the process.
- Process needs to cater for **changing requirements**, whether this be agile software development or traditional change management processes.
- Larger teams need **greater coordination** (team mgmt. and software code/artefact mgmt.) and communication mgmt. (e.g. face-to-face, documents, teleconferences)

Transitioning from Uni to Workplace (4)

Commercial S/W Development:

- Process needs to ensure adequate **team interaction/communication**. This is detail-oriented: versions and patch levels, requirements detail.
- The process must **accommodate and resolve reported customer issues**, this means that we need to identify the exact version of the code / affected artefacts and reproduce them, and fix issues and ship fixes.
- Commercial systems may need greater **precision in design and implementation** – security, reliability and formality of technique.

Transitioning from Uni to Workplace (5)

Commercial S/W Development:

- Process needs to **support customer interaction**, helpdesks, contact numbers, pagers, 24*7*365, support and maintenance teams.
- **Products / Services** must be deployable to **operational platforms** and **run continuously** (or as required).
- Since software **products and services** need to be **maintained and adapted**, the development process must be managed so as to produce software products that are maintainable/adaptable, this includes activities such as architectural design, refactoring, documentation, training, pair programming.

Transitioning from Uni to Workplace (6)

Commercial S/W Development:

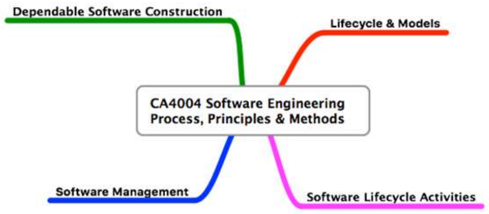
- The process should **guard against elevated levels of tacit knowledge**, otherwise if a person leaves all their knowledge goes with them and we may end up with an economically unviable support/development proposition. Reviewing, training, communications, rotation of work in the process all help to guard against this.
- Since the technology changes in the medium-long term, the process must **support technology shifts**. In effect, the process itself must have mechanisms that allow it to change.

Bottom Line

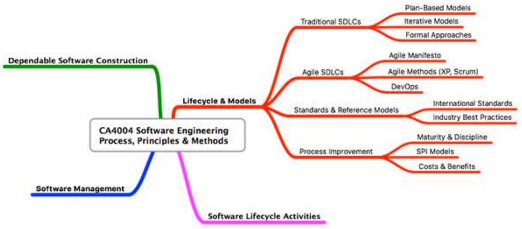
Commercial Software Development is COMPLICATED – manifestly more so than university-based student assignments.

Commercial Software Development is layered in process, whether that be formal, informal or automated in supporting tooling.

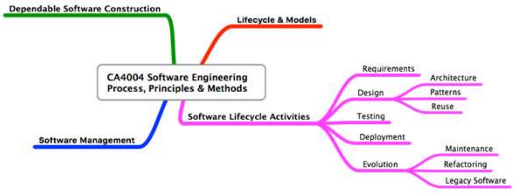
What are we going to study?


Dependable Software Construction
Lifecycle & Models
CA4004 Software Engineering Process, Principles & Methods
Software Management
Software Lifecycle Activities

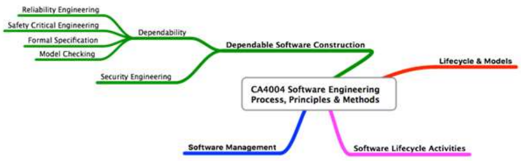
What are we going to study?



What are we going to study?

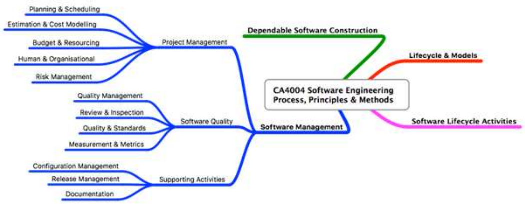


What are we going to study?



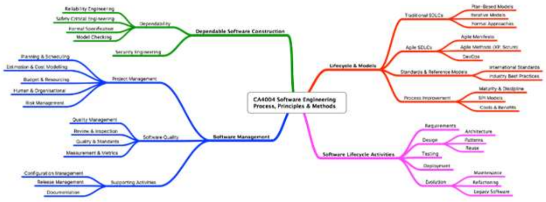
Reliability Engineering
Safety Critical Engineering
Formal Specification
Model Checking
Dependability
Dependable Software Construction
Security Engineering
Lifecycle & Models
CA4004 Software Engineering
Process, Principles & Methods
Software Management
Software Lifecycle Activities

What are we going to study?



Planning & Scheduling
Estimation & Cost Modelling
Budget & Resourcing
Human & Organisational
Risk Management
Project Management
Dependable Software Construction
Lifecycle & Models
CA4004 Software Engineering
Process, Principles & Methods
Quality Management
Review & Inspection
Quality & Standards
Measurement & Metrics
Software Quality
Software Management
Software Lifecycle Activities
Configuration Management
Release Management
Documentation
Supporting Activities

The “Big Picture”



Reliability Engineering
Safety Critical Engineering
Formal Specification
Model Checking
Dependability
Dependable Software Construction
Security Engineering
Planning & Scheduling
Estimation & Cost Modelling
Budget & Resourcing
Human & Organisational
Risk Management
Project Management
Quality Management
Review & Inspection
Quality & Standards
Measurement & Metrics
Software Quality
Software Management
Configuration Management
Release Management
Documentation
Supporting Activities
CA4004 Software Engineering
Process, Principles & Methods
Lifecycle & Models
Traditional SDLCs
Plan-Based Models
Iterative Models
Formal Approaches
Agile SDLCs
Agile Manifesto
Agile Methods (XP, Scrum)
DevOps
Standards & Reference Models
International Standards
Industry Best Practices
Process Improvement
Maturity & Discipline
SPI Models
Costs & Benefits
Software Lifecycle Activities
Requirements
Design
Architecture
Patterns
Reuse
Testing
Deployment
Evolution
Maintenance
Refactoring
Legacy Software

Any questions, before we start the real stuff?

Software Process – A “slightly” humorous perspective

Some background and context information to help you understand what SP and SPI is all about…

In the Beginning…

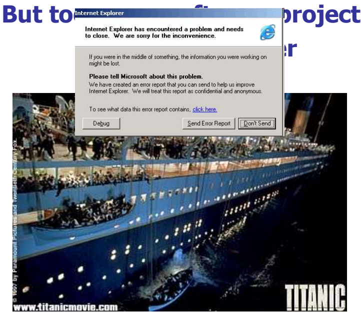
Software projects start with optimism, and sometimes even wild enthusiasm

At the End…

But too many software projects still end is disaster

At the End…

But to … projects … r


Internet Explorer

Internet Explorer has encountered a problem and needs to close. We are sorry for the inconvenience.

If you were in the middle of something, the information you were working on might be lost.

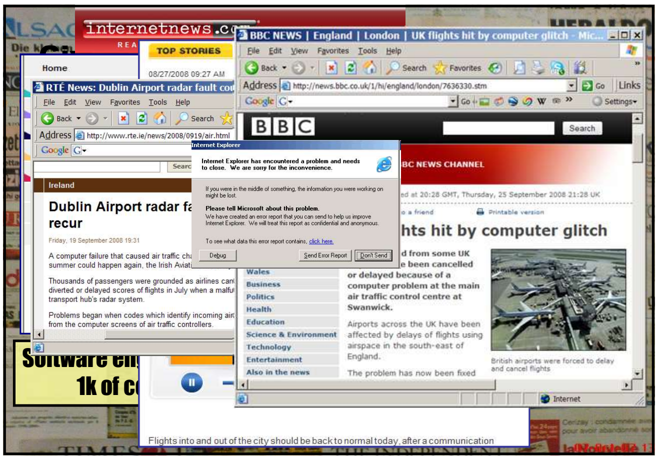
Please tell Microsoft about this problem.

We have created an error report that you can send to help us improve Internet Explorer. We will treat this report as confidential and anonymous.

To see what data this error report contains, click here.

The Blame Game

Many problems are related to management practices



RTÉ News: Dublin Airport radar fault cou

File Edit View Favorites Tools Help

Address http://www.rte.ie/news/2008/0919/air.html

Ireland

Dublin Airport radar fa recur

Friday, 19 September 2008 19:31

A computer failure that caused air traffic cha summer could happen again, the Irish Aviati

Thousands of passengers were grounded as airlines cant diverted or delayed scores of flights in July when a malfu transport hub's radar system.

Problems began when codes which identify incoming air from the computer screens of air traffic controllers.

BBC NEWS | England | London | UK flights hit by computer glitch - Mic...

File Edit View Favorites Tools Help

Address http://news.bbc.co.uk/1/hi/england/london/7636330.stm

BC NEWS CHANNEL

ed at 20:28 GMT, Thursday, 25 September 2008 21:28 UK

hts hit by computer glitch

d from some UK e been cancelled or delayed because of a computer problem at the main air traffic control centre at Swanwick.

Airports across the UK have been affected by delays of flights using airspace in the south-east of England.

The problem has now been fixed

British airports were forced to delay and cancel flights

Internet Explorer

Internet Explorer has encountered a problem and needs to close. We are sorry for the inconvenience.

If you were in the middle of something, the information you were working on might be lost.

Please tell Microsoft about this problem.

We have created an error report that you can send to help us improve Internet Explorer. We will treat this report as confidential and anonymous.

To see what data this error report contains, click here.

Flights into and out of the city should be back to normal today, after a communication

Software Failures List (start from recent):
from SoftwareEngineeringReferences.com

***Software problems in the automated baggage sorting system of a major airport in** February 2008 prevented thousands of passengers from checking baggage for their flights. It was reported that the breakdown occurred during a software upgrade, despite pre-testing of the software. The system continued to have problems in subsequent months.

***News reports in** December of 2007 indicated that significant software problems were continuing to occur in a new ERP payroll system for a large urban school system. It was believed that more than one third of employees had received incorrect paychecks at various times since the new system went live the preceding January, resulting in overpayments of $53 million, as well as underpayments. An employees' union brought a lawsuit against the school system, the cost of the ERP system was expected to rise by 40**%, and the non-payroll part of the ERP system was delayed. Inadequate testing reportedly contributed to the problems.**

***In** November of 2007 a regional government reportedly brought a multi-million dollar lawsuit. The vendor also sued its subcontractor on the project against a software services vendor, claiming that the vendor 'minimized quality' in delivering software for a large criminal justice information system and the system did not meet requirements

*In June of 2007 news reports claimed that software flaws in a popular online stock-picking contest could be used to gain an unfair advantage in pursuit of the game's large cash prizes. Outside investigators were called in and in July the contest winner was announced. Reportedly the winner had previously been in 6th place, indicating that the top 5 contestants may have been disqualified.

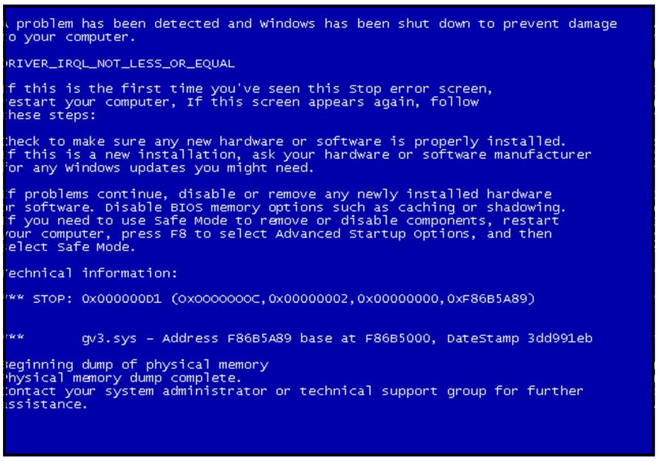
***A** software problem contributed to a rail car fire in a major underground metro system in April of 2007 according to newspaper accounts. The software reportedly failed to perform as expected in detecting and preventing excess power usage in equipment on a new passenger rail car, resulting in overheating and fire in the rail car, and evacuation and shutdown of part of the system.

*Tens of thousands of medical devices were recalled in March of 2007 to correct a software bug. According to news reports, the software would not reliably indicate when available power to the device was too low.

***A** September 2006 news report indicated problems with software utilized in a state government's primary election, resulting in periodic unexpected rebooting of voter checkin machines, which were separate from the electronic voting machines, and resulted in confusion and delays at voting sites. The problem was reportedly due to insufficient testing.

***In** August of 2006 a **U.S. government student loan service erroneously made public the** personal data of as many as 21,000 borrowers on it's web site, due to a software error. The bug was fixed and the government department subsequently offered to arrange for free credit monitoring services for those affected.

***A software error reportedly resulted in** overbilling of up to several thousand dollars to each of 11,000 customers of a major


```
 problem has been detected and windows has been shut down to prevent damage
o your computer.

RIVER_IRQL_NOT_LESS_OR_EQUAL

f this is the first time you've seen this Stop error screen,
estart your computer. If this screen appears again, follow
hese steps:

heck to make sure any new hardware or software is properly installed.
f this is a new installation, ask your hardware or software manufacturer
or any Windows updates you might need.

f problems continue, disable or remove any newly installed hardware
r software. Disable BIOS memory options such as caching or shadowing.
f you need to use Safe Mode to remove or disable components, restart
our computer, press F8 to select Advanced Startup Options, and then
elect Safe Mode.

echnical information:

** STOP: 0x000000D1 (0x0000000C,0x00000002,0x00000000,0xF86B5A89)

**       gv3.sys - Address F86B5A89 base at F86B5000, DateStamp 3dd991eb

eginning dump of physical memory
hysical memory dump complete.
ontact your system administrator or technical support group for further
ssistance.
```

The Industry Is Challenged

Project failure statistics are scary…

Software delivery remains an art to some extent, not pure science.

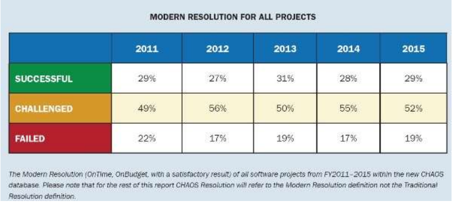
It has been consistently difficult to manage delivery of software in a predictable and reliable manner!

Software Developers need to:

Increase the predictability of quality delivery, on time and within budget

The Industry Is Challenged

Project failure statistics are scary…



MODERN RESOLUTION FOR ALL PROJECTS

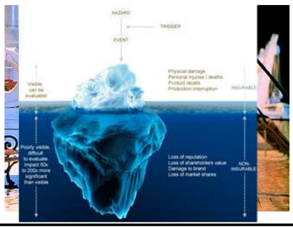
	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011–2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

About Icebergs

One could imagine that the passengers standing on the deck of Titanic might have admired the icebergs they saw as exotic phenomena

You would also think that most of them probably had no concept of the anatomy of an iceberg



I will tell you a secret

90% of an iceberg lies underwater

This is not a secret

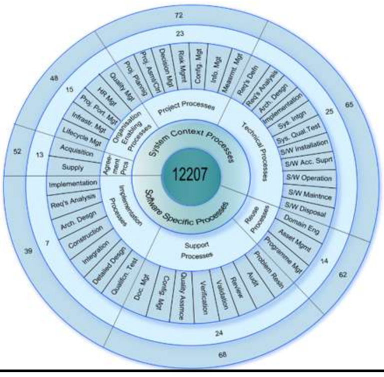
What is not usually appreciated is that software systems are like that too – there is a user interface that can consume a (very) small proportion of the work, with much of the development effort and thus the cost hidden from view.

This is not a total secret

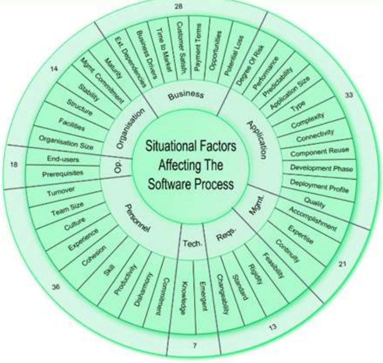
The secret is…

people outside the business often do not understand this (and some inside the business may not understand it well enough either!)

One of the issues: Complexity(1)



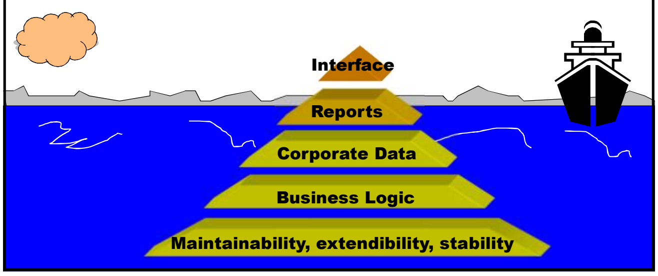
One of the issues: Complexity(2)



One of the issues: Complexity (3)

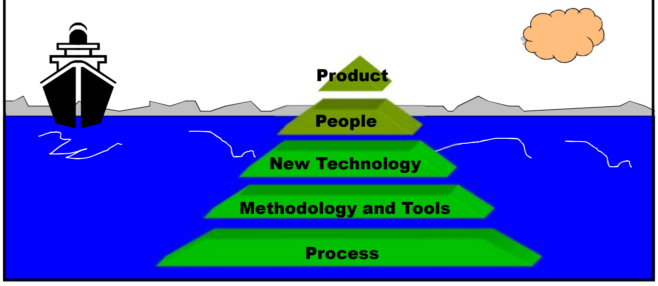
Software Products

We do not pay enough attention to what lies beneath the surface



Software Projects

We do not pay enough attention to what lies beneath the surface

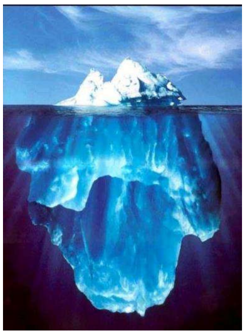


The Tip of the Iceberg

We do not pay enough attention to what lies beneath the surface

Blame it on…

- Computer Programmers
- The Web
- Impatience of customers
- etc…



The Challenge

Change perspective and understanding

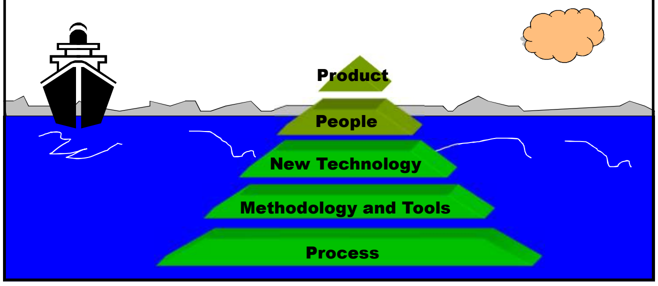
As a Software Community we are still challenged with the task to, in a pedagogic way, create enough understanding for our work

So we may have the time and resources that we need in order to deliver the quality in the software that the customer expects

That is, we need to convey that it is not only the surface (tip of the Iceberg) that matters

Software Projects

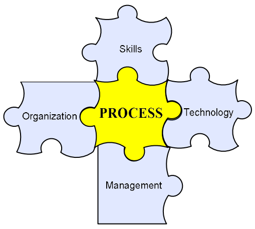
We do not pay enough attention to what lies beneath the surface



Quality Process Thinking

A common process that everyone adheres to has benefits:

- Alignment of group activities towards achieving the **common goals**
- Proving the basis for objective **measurement**
- Enhancing **consistency** and **repeatability** of the process results and group performance
- Improving the overall **business performance**

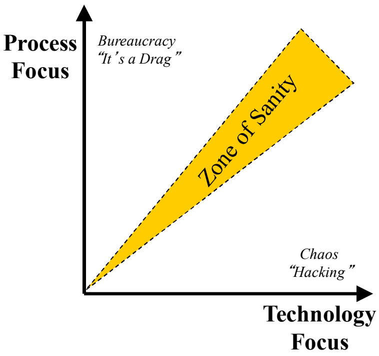


> *Processes are like habits: hard to establish and even harder to break (Humphrey 1995)*

The Cure?

Is having a
[Quality]
Software Process
the cure for everything?

The Process Dilemma – a case for Balance



For Modern Organisations

There is evidence to show that a **[Quality] Software Development Process** has a significant return on investment and impact in improving quality while decreasing cycle time and costs

These lessons need to be applied across the enterprise

- That is, improvements should be made in all areas
  - Process, people and technology
  - All engineering disciplines, not just software

The goal is to produce software better, faster and cheaper

Some Quality Process Concepts include...

Project Management

Schedule Planning

Cost Estimation

Risk Management

Coding standards

Testing

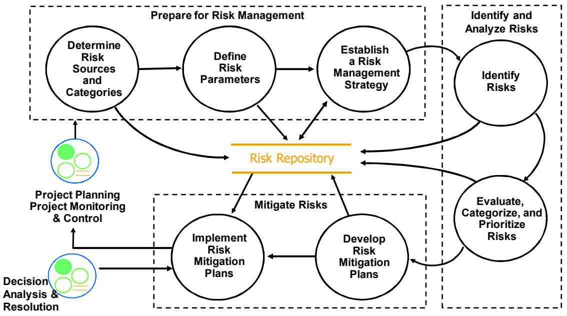
Many others...

Risk Management

Identify potential problems before they occur

So that risk handling activities may be planned and invoked as needed across the life of the product or project to mitigate adverse impacts

A Risk Management Process


Prepare for Risk Management
Determine Risk Sources and Categories
Define Risk Parameters
Establish a Risk Management Strategy
Identify and Analyze Risks
Identify Risks
Risk Repository
Evaluate, Categorize, and Prioritize Risks
Mitigate Risks
Implement Risk Mitigation Plans
Develop Risk Mitigation Plans
Project Planning
Project Monitoring
& Control
Decision Analysis & Resolution

CMMI Practice-to-Goal relationship for Risk Management

SG 1 Prepare for Risk Management
- SP 1.1-1 Determine Risk Sources and Categories
- SP 1.2-1 Define Risk Parameters
- SP 1.3-1 Establish a Risk Management Strategy

SG 2 Identify and Analyze Risks
- SP 2.1-1 Identify Risks
- SP 2.2-1 Evaluate, Categorize, and Prioritize Risks

SG 3 Mitigate Risks
- SP 3.1-1 Develop Risk Mitigation Plans
- SP 3.2-1 Implement Risk Mitigation Plans

In CMMI guide there are 22 pages describing Risk Management and its associated activities

GG 1 Achieve Specific Goals
- GP 1.1 Perform Base Practices

GG 2 Institutionalize a Managed Process
- GP 2.1 Establish an Organizational Policy
- GP 2.2 Plan the Process
- GP 2.3 Provide Resources
- GP 2.4 Assign Responsibility
- GP 2.5 Train People
- GP 2.6 Manage Configurations
- GP 2.7 Identify and Involve Relevant Stakeholders
- GP 2.8 Monitor and Control the Process
- GP 2.9 Objectively Evaluate Adherence
- GP 2.10 Review Status with Higher Level Management

GG 3 Institutionalize a Defined Process
- 3.1 Establish a Defined Process
- GP 3.2 Collect Improvement Information

GG 4 Institutionalize a Quantitatively Managed Process
- GP 4.1 Establish Quantitative Objectives for the Process
- GP 4.2 Stabilize Subprocess Performance

GG 5 Institutionalize an Optimizing Process
- GP 5.1 Ensure Continuous Process Improvement

TO CONCLUDE

Conclusion

- Commercial Software Development is much more COMPLICATED than university-based student assignments.
- Commercial Software Development is layered in process, whether that be formal, informal or automated in supporting tooling.
- Commercial Software Development processes are highly variable dependent on situational context.