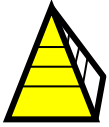


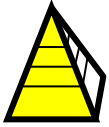
# Why Document-centric Development Life Cycles Cause Problems

F. Alan Goodman – “The Process Guy”



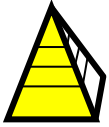
# What I Cover...

- The defect problem when mixing-and-matching work products versus documents in the development life cycle
- My process framework architecture that separates these notions
- Software development operations that support these notions



# Overview

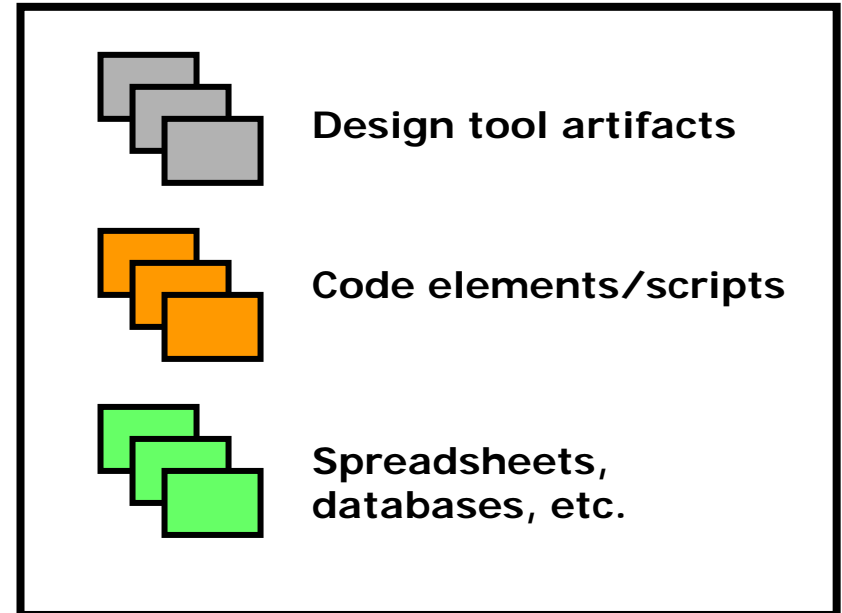
- Defects can be introduced into the software development life cycle by not understanding the difference between “work product documentation development” and “document development”
- You need a process framework architecture that supports this difference!
- Many companies mix-and-match these with disastrous results!



# Developers' World of Work Products

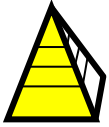


Versions  
↔



**Systems/Software Engineers**

**Engineers like computer science related stuff and software tools,  
not writing documents! That's their training and expertise.**



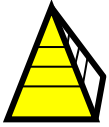
# Developers typically do not understand...



~~Company document standards  
Header/footer formatting  
Document revision numbering  
Document formatting  
Document fonts/styles  
Document script usage  
Document spacing  
Document lists  
Document graphics/text layouts  
Document-based software  
etc...~~

**Systems/Software Engineers**

**Engineers have no training and expertise in this area – and it shows!**



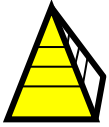
# System/Software Developers

Real world indicators:

- Developers like to design/program only (that's why they have CS degrees)
- They do not like documentation
- They are not English Majors
- They are not good at grammar and spelling
- They don't understand documentation standards
- Many programmers today have English as their second language

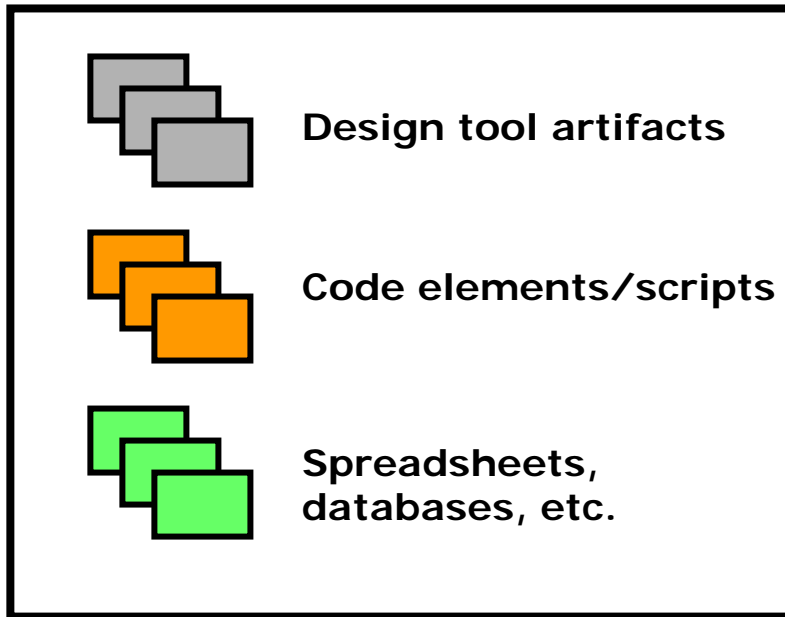
Given these real world indicators:

- Restrict developers to work products, tools, etc. (their world)
- Never have developers produce documents – have them produce the input grist to documents.

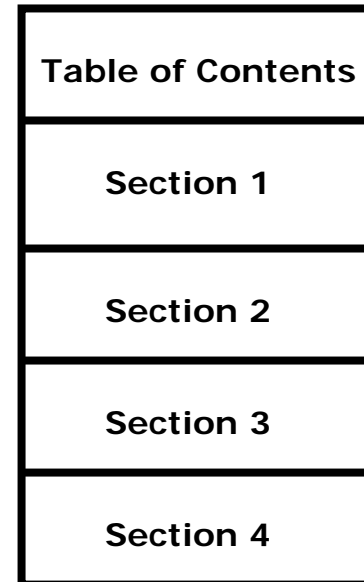


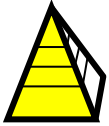
# Distribution of Labor

**Work products produced  
by system/software engineers**



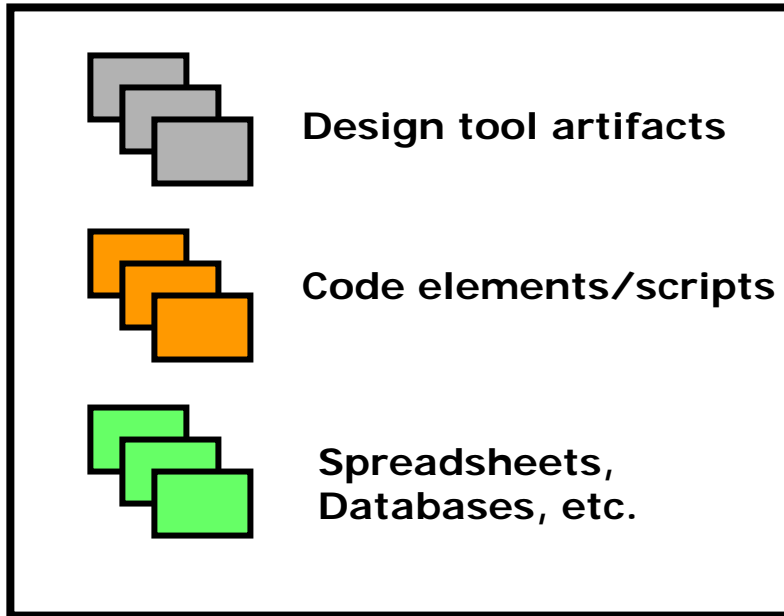
**Document produced  
by Technical Documentation**





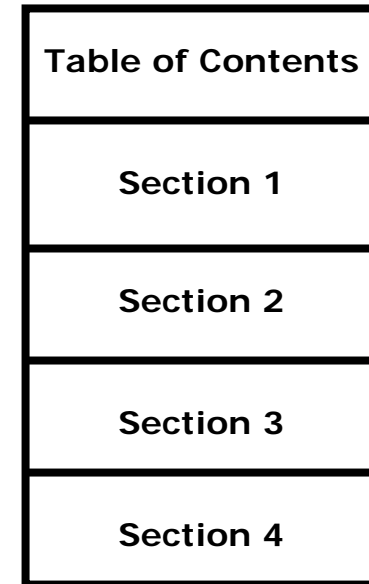
# Work Product Documentation vs. Documents

## Work products

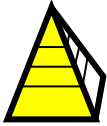


**Very dynamic by the minute**  
**No formatting**

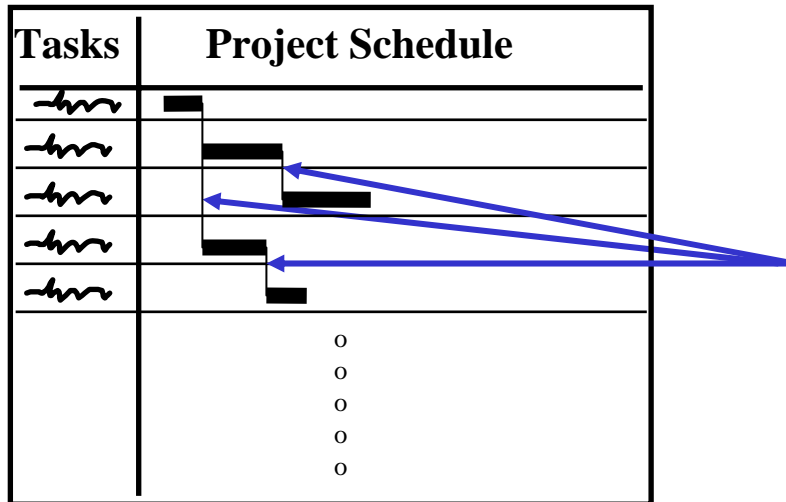
## Document



**Not dynamic – a snapshot**  
**formatted**

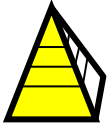


# Potential Problem in a Nutshell

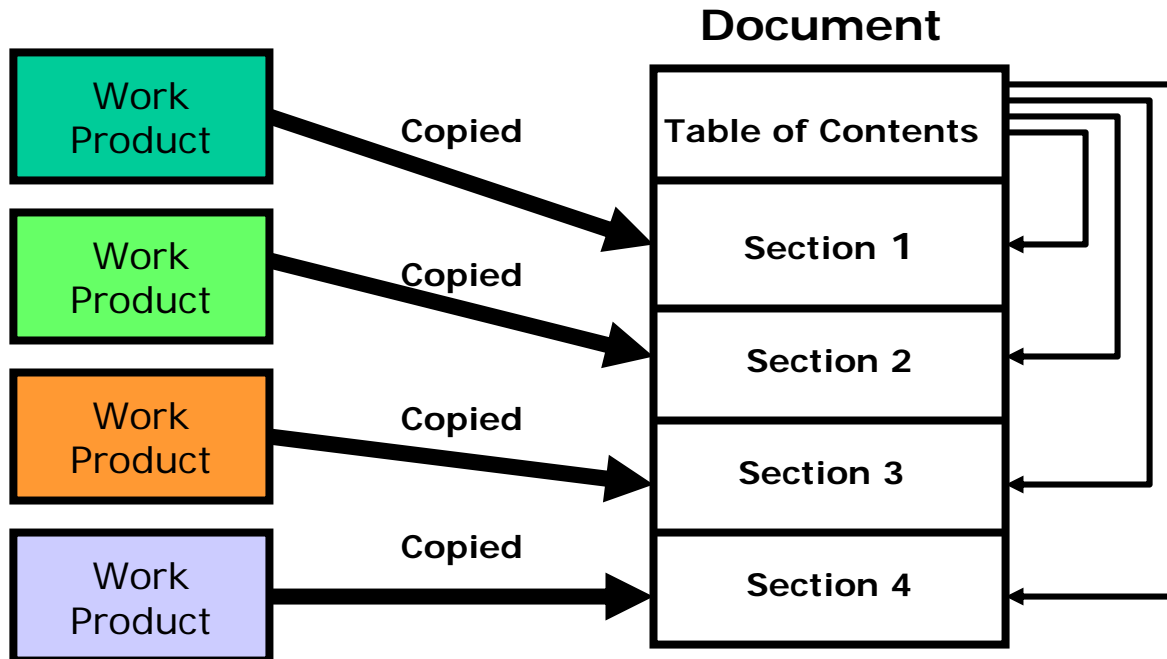


- Software project schedules are a series of interconnecting (predecessor/successor) tasks.
- Each task produces one (or more) work products for “done-ness.”
- Work products should be version-controlled for potential rework and control.
- Work products may use a variety of tool sets and software applications.
- Work products are, by definition, only a piece of the whole story and dynamic.
- Work products are created by designers/developers.

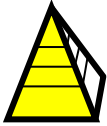
**Engineering must NEVER produce a document and treat it like a work product.**



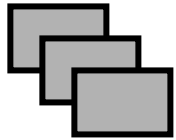
# Think about many documents ...



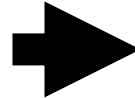
**Once done, you now have TWO data sources,  
a fundamental data base integrity issue.**



# The Story of Two Development Leads



Design tool artifacts



## Lead #1

- Design comes from the latest version of the tool work product
- Design is the correct design basis

Table of Contents
Section 1
Section 2
Section 3
Section 4

Design document



## Lead #2

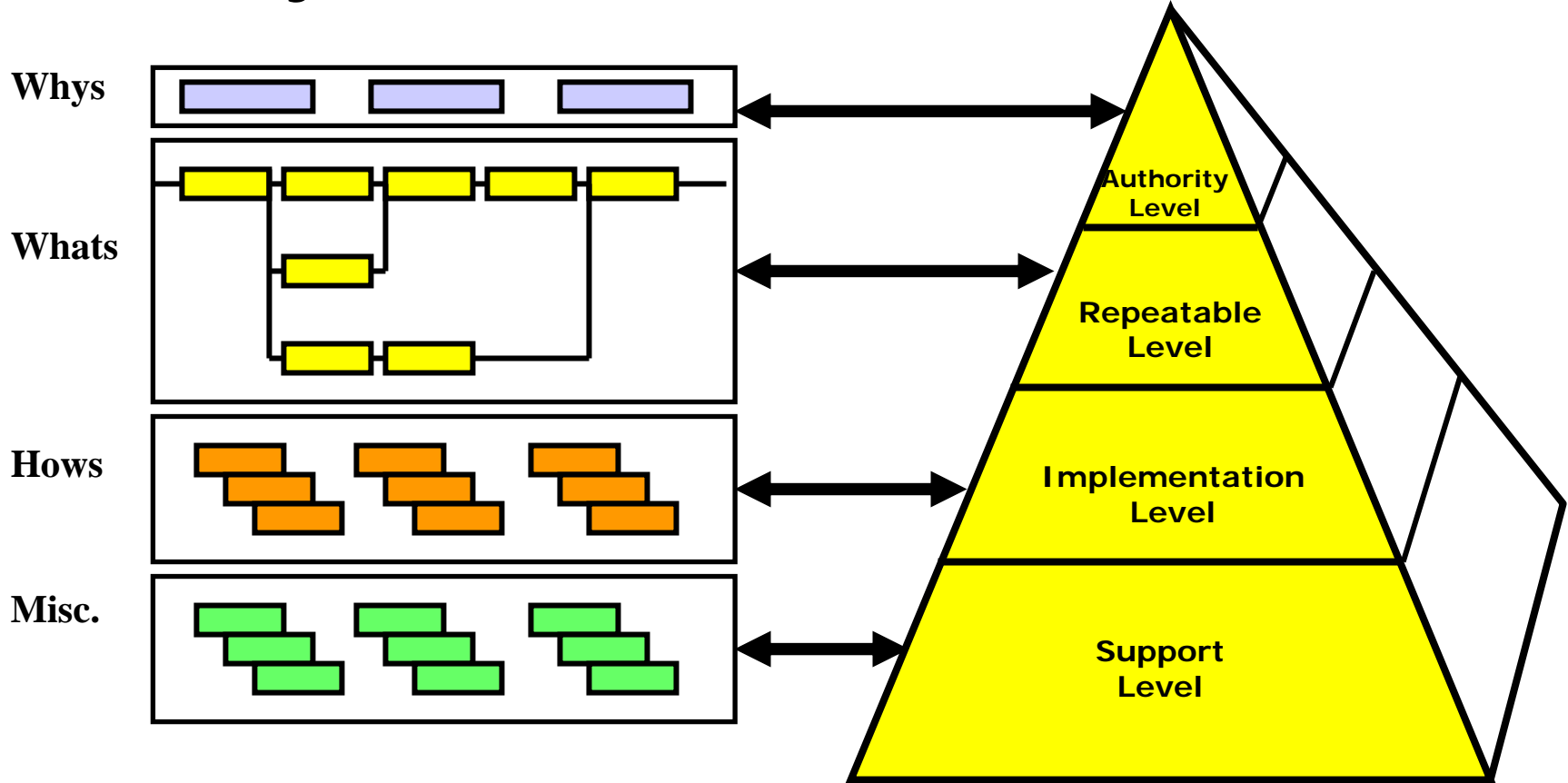
- Design comes from design document
- Design in tool work product has moved on
- Design is not correct as a basis?????

**\*\*\*\*DEFECT\*\*\*\***

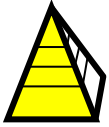
**This happens all too often without a solid process foundation structure enforcing this separation!**



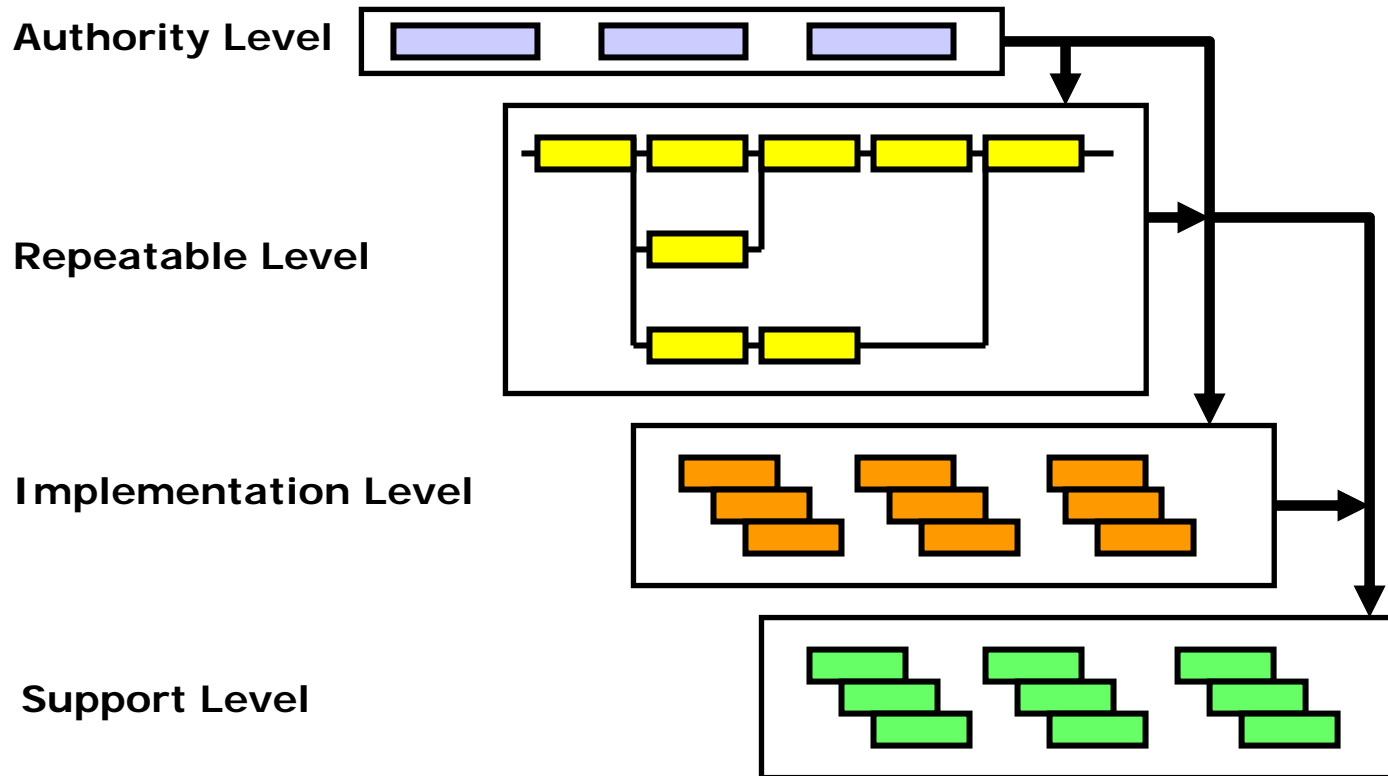
# My Process Framework Architecture



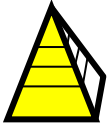
**An organized structure that supports real world operations**



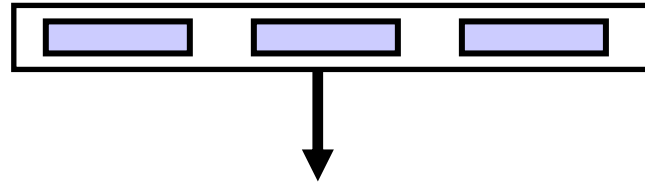
# Connecting the Process Layers



**An organized structure that supports real world developmental operations**



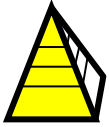
# Authority Level



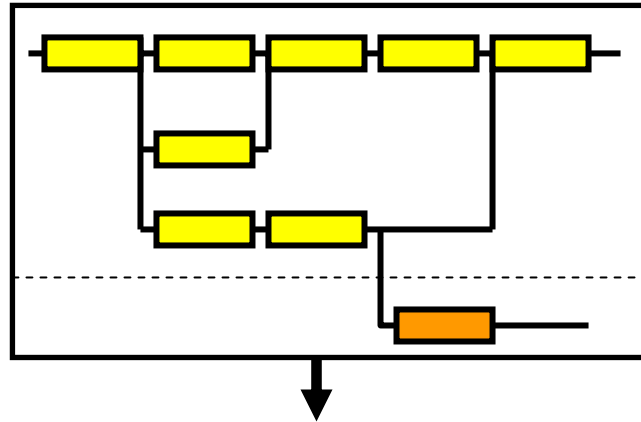
Has high-level “what” requirements for compliance:

- Maturity models, such as CMMI /Six Sigma, ...
- International standards, such as ISO 9001, IEEE, ...
- Governmental regulations, such as SOX, FDA, FAA, ...
- Company policies

**These are entities that you want compliance matrices for –  
to address the existence part of compliance.  
Interest only to SEPG – not practitioners!**

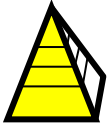


# Repeatable Level

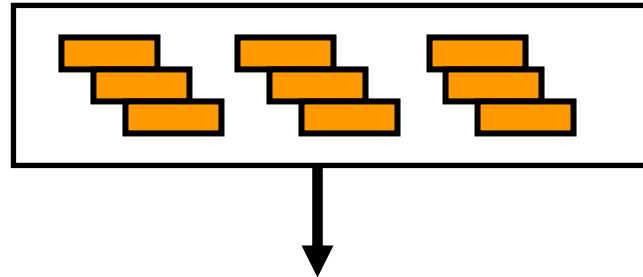


- Instances show up as project schedule tasks
- Provides project management/developmental roadmap
- Accessed using the intranet or project schedules
- Connected to “how-to” elaborations/tools (Implementation Level)
- Defines steps/inputs/outputs roles/training/metrics, etc.
- Portable across divisions. Can port to subcontractors.

**All project schedule tasks are directly connected to this level for scheduled work.**

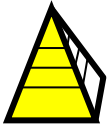


# Implementation Level

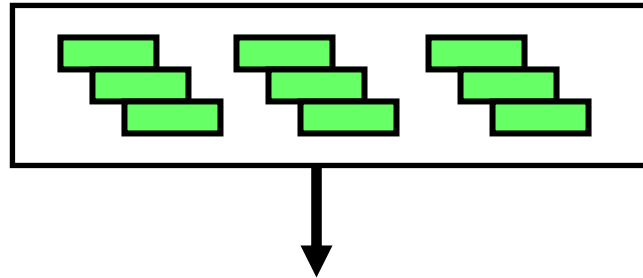


- How-to procedural level
- Divided into scheduled/non-scheduled procedures/events
  - Scheduled procedures connected to the schedule tasks
  - Non-scheduled events connected to phases in the life cycle
- Supports extensibility/flexibility for how-to options
- Supports scalability/cross-site/tool variations

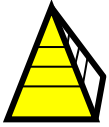
**Nothing exists here unless connected to a “what” at a higher level**



# Support Level



- **Work product templates, inspections checklists, placeholders, examples**
- **Form templates, inspection checklists, placeholders, examples**



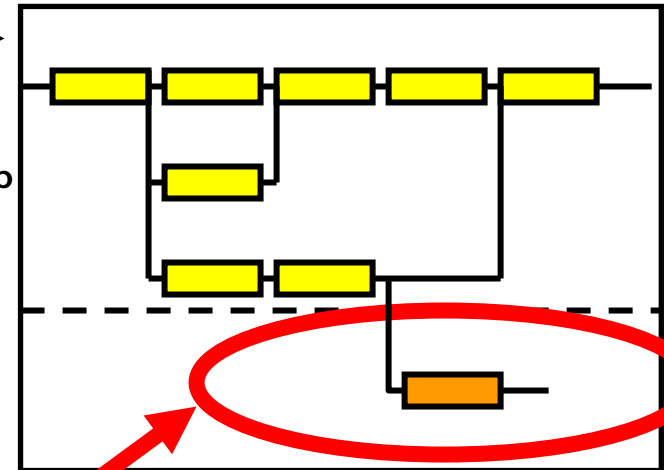
# Repeatable Processes and the Schedule Should Be ....

## Activity Instances on a Schedule

Tasks	Project Schedule
	○
	○
	○
	○
	○

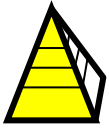
→ Collapse the activity instances to get the end-to-end process map

## Process Activities

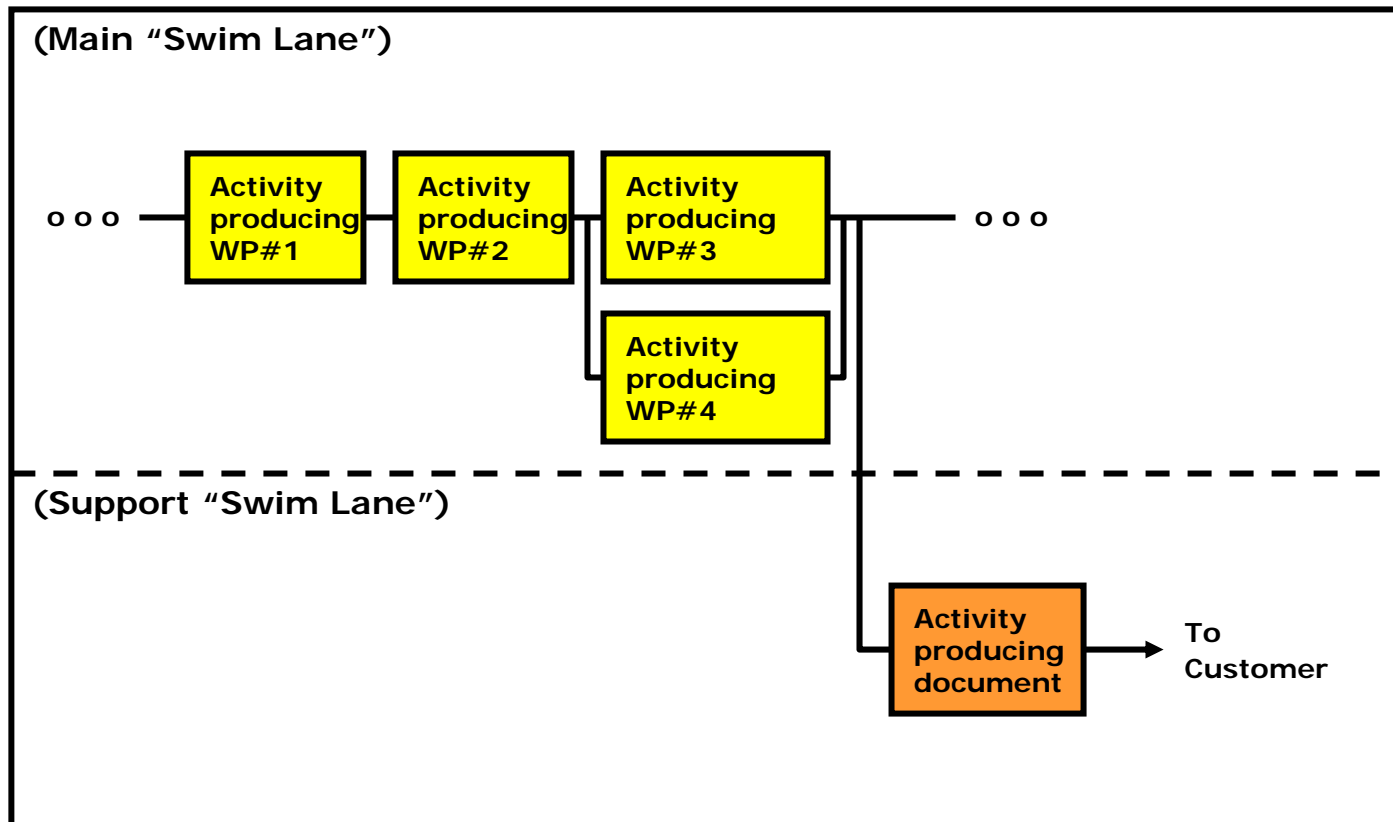


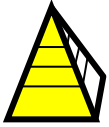
← Create instances from the activity map to get the project schedule

**Where document production fits in**



# Documents from a Process Perspective





# Types of Delivered Documents

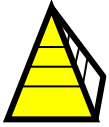
## Real documents

- Gather work products, refines, edits
- Extra verbiage “glue” added
- “Pretty print” formatting/presentation
- Compliance to a document standard
- Single deliverable entity typically
- Higher cost

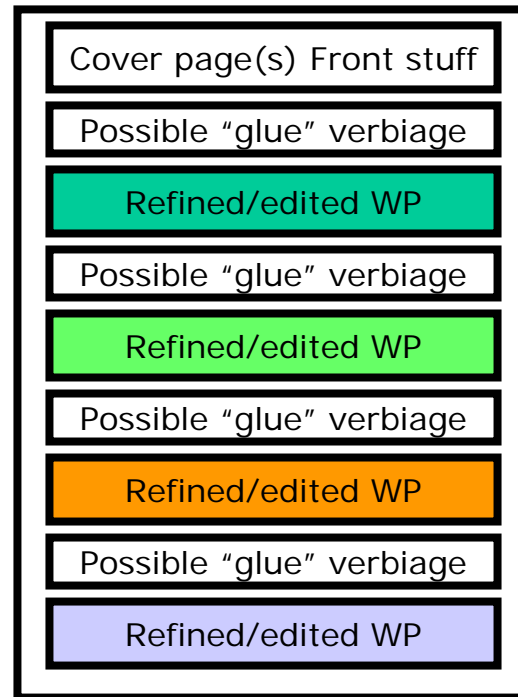
## Virtual documents

- Reference or directly include work products as separate files
- Little to no extra verbiage “glue” (cover for control)
- No document standard involved, except for contract cover page(s)
- Single to multiple deliverable entity or entities
- Lower cost

**Allow customers a price option. Both can be achieved using process selection.**



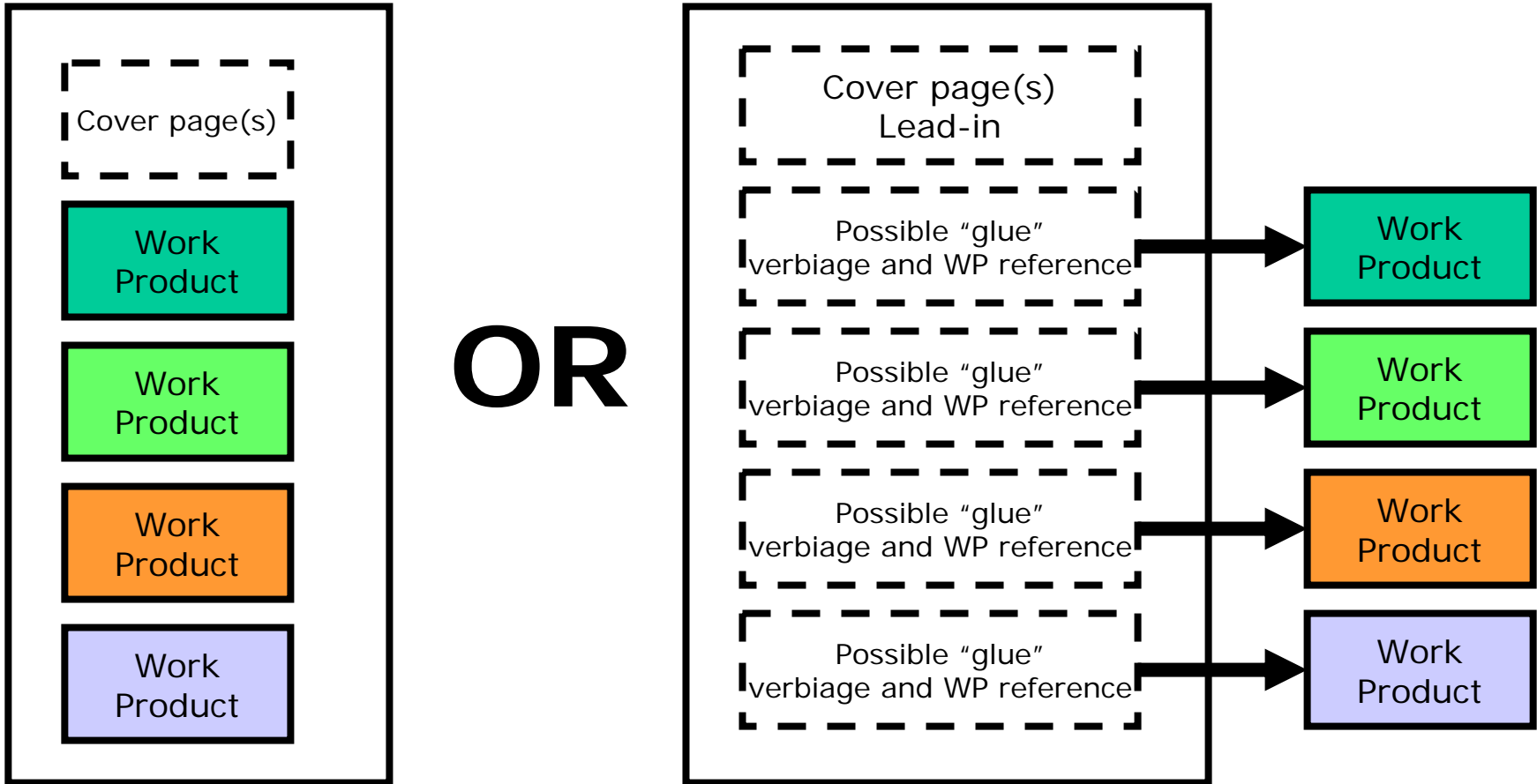
# Real Documents

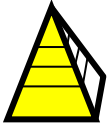


**Work products physically inserted into the end document either as-is or edited/refined. "Glue" verbiage added.**



# Virtual Documents

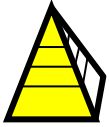




# Price Document Deliverables Separately

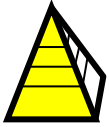
- Process execution ALWAYS produces work products.
- Work products CAN be used in lieu of packaged documents as deliverables (virtual documents).
- If customer wants packaged work products as documents, price accordingly (real documents).
- Separate document production support activities from main engineering activities.

**Process facilitates both real and virtual documents.  
Customers choose which they want!**



# Using Separate Document Activities

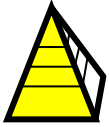
- You can cost/price document production easily
- You can base future estimations on past actuals
- You can separate out from main engineering activities
- You can get metrics
  - Percentage of project costs for document preparation
  - Percentage of support costs for document preparation
  - Cost per type of document
  - Etc.



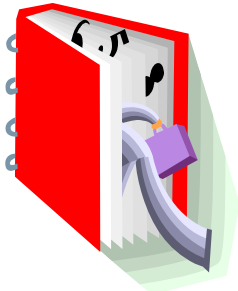
# Summary

- Never, never use documents within the development life cycle as work products! Can create defects!
- Never, never mismatch role expertise within the development life cycle:
  - Engineering does engineering stuff
  - Technical documenters do document stuff
- Separate out document preparation and production as support activities for scheduling

**The process foundation organization itself can really help enforce these concepts.**

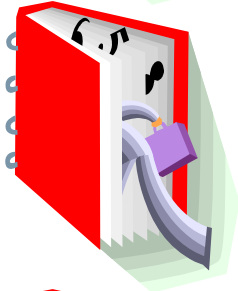


# Connection to My Books



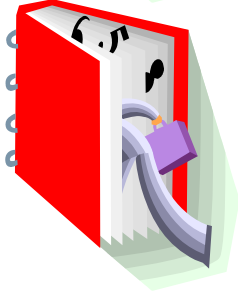
**"Defining and Deploying  
Software Processes"**  
ISBN # 0-8493-9845-2

Describes my process framework architecture model along with people issues to insert in a company culture



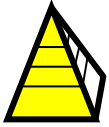
**"Process-Based Software  
Project Management"**  
ISBN # 0-8493-7304-2

Describes enormous benefits for software project management using my process framework architecture. (UCSD Extension class of the same name)



**"Process-Based Software  
Development  
-- for Operational Success"  
(working title)**  
ISBN # 0-8493-xxxx-x

Describes a step-by-step guide to actually achieving this within a company (work-in-progress)



**Thank you for your attention.**