SMC-TR-05-21

AEROSPACE REPORT NO.
TR-2005(8550)-2

Evolutionary Acquisition and Spiral Development Tutorial

30 September 2005

Prepared by

P. HANTOS
Software Acquisition and Process Department
Software Engineering Subdivision

Prepared for

SPACE AND MISSILE SYSTEMS CENTER
AIR FORCE SPACE COMMAND
2430 E. El Segundo Boulevard
Los Angeles Air Force Base, CA 90245

Engineering and Technology Group

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

THE AEROSPACE CORPORATION
El Segundo, California

This report was submitted by The Aerospace Corporation, El Segundo, CA 90245-4691, under Contract No. FA8802-04-C-0001 with the Space and Missile Systems Center, 2430 E. El Segundo Blvd., Los Angeles Air Force Base, CA 90245. It was reviewed and approved for The Aerospace Corporation by Mary A. Rich, Principal Director, Software Engineering Subdivision, Computer Systems Division. Michael Zambrana was the project officer for the Mission-Oriented Investigation and Experimentation (MOIE) program.

This report has been reviewed by the Public Affairs Office (PAS) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication. Publication of this report does not constitute Air Force approval of the report's findings or conclusions. It is published only for the exchange and stimulation of ideas.

Michael Zambrana
SMC/AXE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (*DD-MM-YYYY*)	2. REPORT TYPE	3. DATES COVERED *(From - To)*
30/09/05		

4. TITLE AND SUBTITLE

Evolutionary Acquisition and Spiral Development Tutorial

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

P. Hantos

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

The Aerospace Corporation
El Segundo, CA 90245-4691

8. PERFORMING ORGANIZATION REPORT NUMBER

TR-2005(8550)-2

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Space and Missile Systems Center
Air Force Space Command
2450 E. El Segundo Blvd.
Los Angeles Air Force Base, CA 90245

10. SPONSOR/MONITOR'S ACRONYM(S)

SMC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

SMC-TR-05-21

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The publication of Department of Defense (DOD) Directives 5000.1 and 5000.2 established a preference for the use of evolutionary acquisition and spiral development in the acquisition of complex weapon systems. NSS Acquisition Policy 03-01 provided some space-oriented customization and, similarly to the original DOD directives, also positioned Evolutionary Acquisition and Spiral Development as preferred strategies for the space domain. The tutorial's key objectives are to establish proper life cycle modeling foundations, and to emphasize that these approaches should be implemented not simply for policy compliance's sake, but because they represent a prudent risk mitigation strategy. Besides the Spiral Development Model, more development organizations have chosen RUP® as their main development life cycle model. A secondary objective of this presentation is to explain the similarities and differences between Spiral Development and RUP®, and provide tangible life cycle modeling guidelines for the acquisition organizations during the formative stages of the contracting process.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					P. Hantos
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(include area code)*
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED		105	(310)336-1802

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. 239.18

Acknowledgements

Development of the Evolutionary Acquisition and Spiral Development Tutorial and preparation of this report was funded by the Software Acquisition task of the Mission-Oriented Investigative Experimentation (MOIE) research project.

This work would not have been possible without the following:

Reviewers

- **Richard J. Adams**, Software Acquisition and Process Office

- **Dr. Joe Betser**, Software Engineering Subdivision

- **Suellen Eslinger**, Software Acquisition and Process Office

- **Dr. Leslie J. Holloway**, Software Engineering Subdivision

Sponsor

- Michael Zambrana, USAF Space and Missile Systems Center, Directorate of Systems Engineering

Funding Source

- Mission-Oriented Investigative Experimentation (MOIE) Research Program (Software Acquisition Task)

Inspiration

- Dr. Barry W. Boehm, University of Southern California

Contents

Use of Trademarks, Service Marks and Trade Names

Use of any trademarks in this material is not intended in any way to infringe on the rights of the trademark holder. All trademarks, service marks, and trade names are the property of their respective owners.

This page is intentionally blank

Introduction

Objectives

- Provide an introduction to development and acquisition life cycle models
- Explain the synergy between development and acquisition life cycle models
- Interpret DOD guidelines on Evolutionary Acquisition and Spiral Development
- Interpret Section 803 of Public Law 107-314 as it relates to Evolutionary Acquisition and Spiral Development
- Demonstrate that Evolutionary Acquisition and Spiral Development are prudent risk mitigation strategies
- Provide guidance on risk-based development and acquisition life cycle model selection

Acronyms:
DOD: Department of Defense

Notes:
It is not the intention of this tutorial to:
- Provide a basic introduction to system acquisition
- Provide a detailed acquisition documentation preparation guide

Approach

- **Emphasis on software**
  - Modern weapon systems are highly software-intensive
    - F22 ~ 2000 KLOC, providing 80%(!) of functionality*
    - "For space systems, the software **is** the CONOPS!"**
- **Pattern-based use of Life Cycle Models (LCMs)**
  - Life Cycle Models
    - LCMs are frameworks, providing a common conceptual frame of reference
    - Clarifying relationships, identifying key elements
    - Providing an abstract, simplified view of reality
  - Patterns
    - Patterns represent a perceptual structure, a customary way of operations
    - "To See Is to Understand"

      --- *Keith Devlin, in "Mathematics, The Science of Patterns"*

GSAW 2005 – Peter Hantos 5 THE AEROSPACE CORPORATION

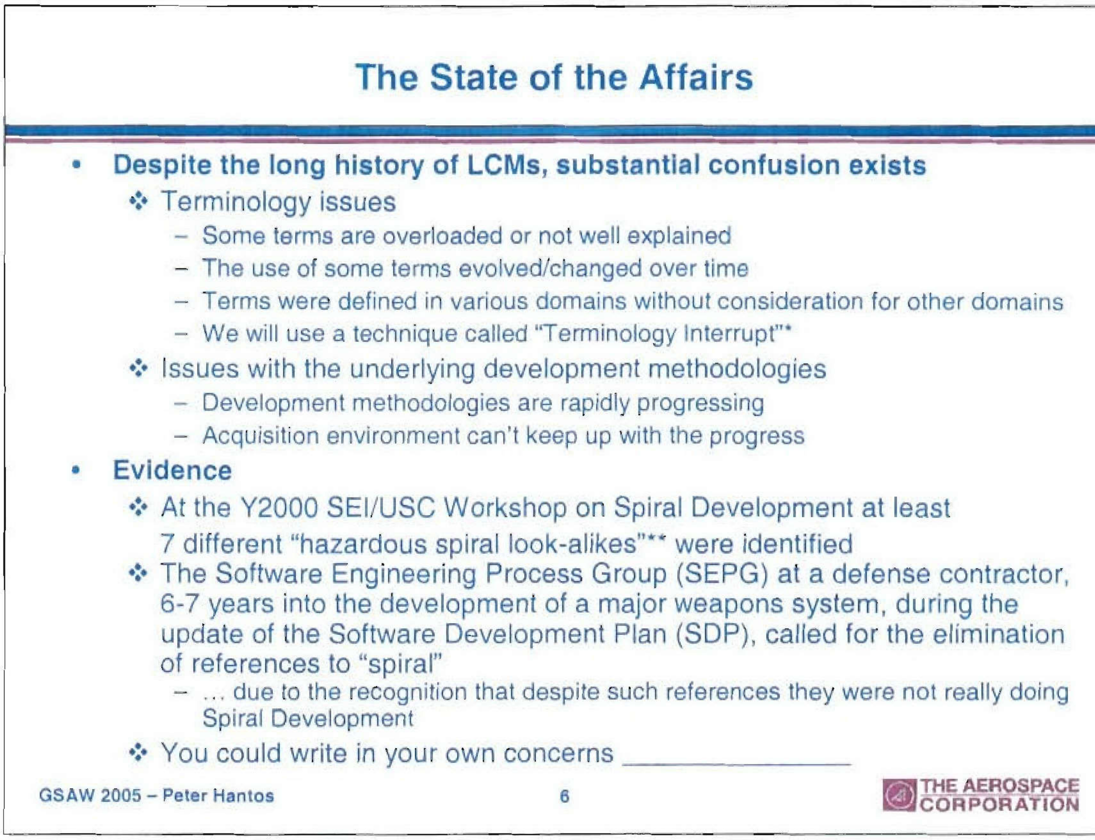
Acronyms:

CONOPS:	Concept of Operations
KLOC:	Thousand Lines of Code
US:	The United States
WBS:	Work Breakdown Structure

Sources Quoted:

* US Air Force Bold Stroke Executive Course, 1992

** Linda Stephenson, Principal Engineer (retired), The Aerospace Corporation



Acronyms:

COTS:	Commercial Off-The-Shelf
LCM:	Life Cycle Model
SEI:	Software Engineering Institute
USC:	University of Southern California

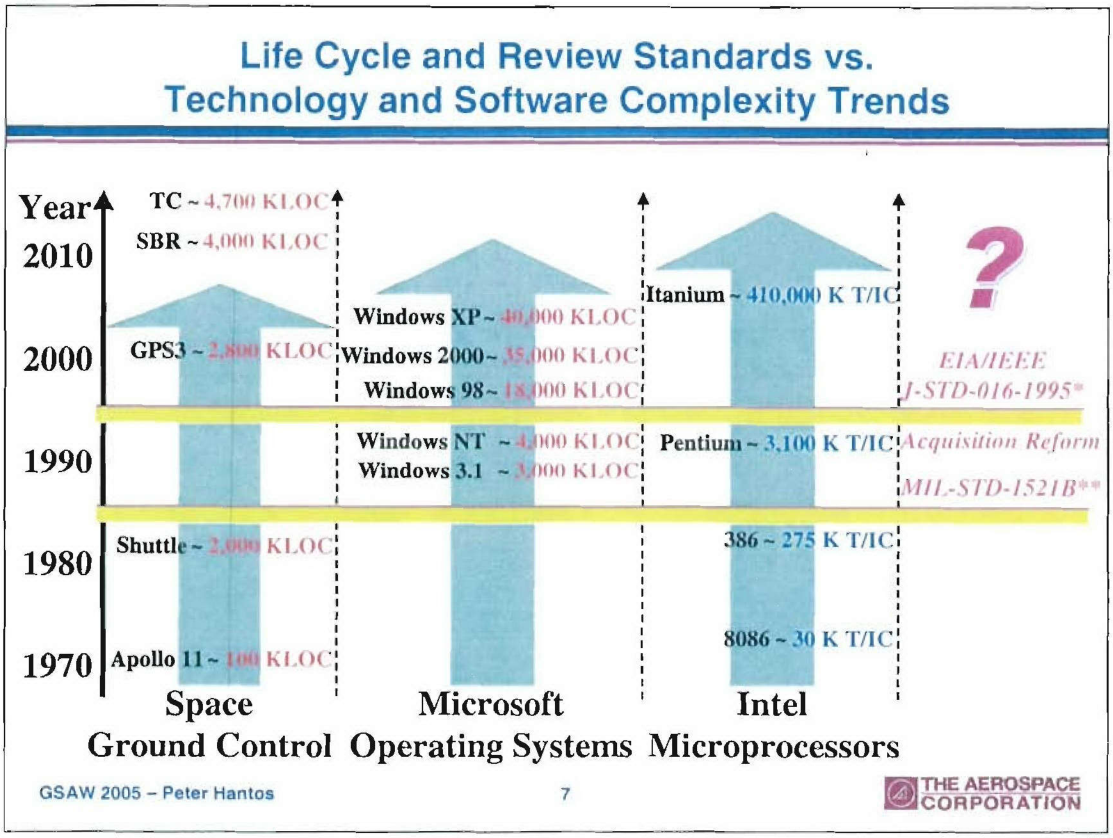
Notes:

***Terminology Interrupt – What is it?**

While many authors choose to explain all definitions and terms either up-front or in an appendix, in this material I follow a strategy I call "Terminology Interrupt." The purpose of this approach is to focus the audience's attention on only those definitions that are needed to understand the immediately following slides. Also, with this technique we can take advantage of the learning process, so during the introduction of new terms we can build on the facts that were already explained in the earlier parts of the material.

** **Hazardous Spiral Look-Alikes [Boehm00]:**

1. Incremental sequential Waterfalls with significant COTS, user interface, or technology risks
2. Sequential spiral phases with key stakeholders excluded from phases
3. Risk-insensitive evolutionary or incremental development
4. Evolutionary development with no life-cycle architecture
5. Insistence on complete specifications for COTS, user interface, or deferred-decision situations
6. Purely logical object-oriented methods with operational, performance, or cost risks
7. Impeccable spiral plan with no commitment to managing risks



<u>Acronyms:</u>

EIA:	Electronics Industry Association
GPS:	Global Positioning System
IEEE:	Institute of Electrical and Electronics Engineers
KLOC:	Thousand Lines of Code
K T/IC:	Thousand Transistors per Integrated Circuit
SBR:	Space Based Radar
TC:	Transformational Communications

<u>Chart Data Sources:</u>
Ground control-related data: Ada Joint Program Office [Ada97] and Steve Burrin [Burrin04]
Microsoft operating systems: David A. Wheeler [Wheel00]
Intel microprocessors: "Silicon – Moore's Law" on the Intel Corporation website [Intel]

<u>Notes:</u>
* **J-STD-016-1995** Standard for Information Technology Software Life Cycle Processes, Software Development, Acquirer-Supplier Agreement: An interim standard, released in September 1995 by the EIA/IEEE
** **MIL-STD-1521B** Military Standard for Technical Reviews and Audits for Systems, Equipments, and Computer Software: Last updated in June 1985(!)

In the 1970s both acquisition and development life cycles were strictly sequential. The prevalent development standard of the times, DOD-STD-2167/DOD-STD-2167A Military Standard for Software Development, was developed to be consistent with primarily a sequential life cycle model, also called the "Waterfall." The related, MIL-STD-1521B standard also structured the technical reviews around the Waterfall milestones.

Successor life cycle standards, like J-STD-016-1995, while they didn't explicitly mandate a sequential development process, still didn't facilitate well the use of more sophisticated development strategies. As the chart shows, system complexity, driven by new developments in hardware/software technologies, has been and is dramatically increasing. At the same time, neither life cycle standards nor review standards have kept up with the incredible pace of progress.

The pressure is on!

This page is intentionally blank

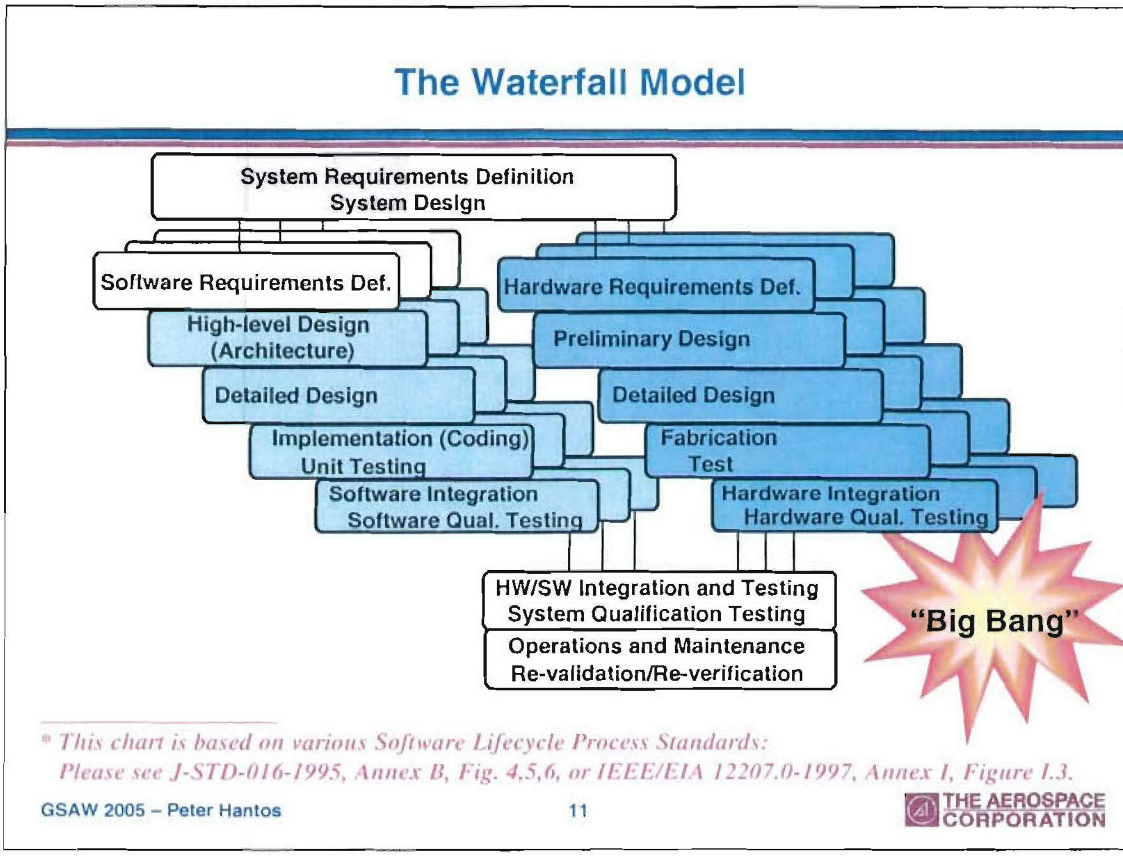
Introduction to Development Life Cycle Models

Life Cycle Modeling Basics

- **A software/system Life Cycle Model is a project management framework**
  - LCMs provide a sequencing strategy; a disciplined approach to structure and document the order of activities
  - LCMs also provide the basis for
    - **Effort and Cost Estimation**
    - **Actual project schedule development**
- **Abstraction in life cycle modeling**
  - LCMs focus on the process aspects of development
  - How are they different from other, related techniques?
    - **Architectural models** focus on the product
    - **Gantt charts** show the actual duration of activities
    - **Work Breakdown Structures** (WBS) define work products and tasks of the development process, but do not provide information on how to make the product
- **We begin the LCM discussion with the models of the development domain**
  - This order is beneficial for instructional purposes
    - It reflects historical trends

GSAW 2005 – Peter Hantos 10 THE AEROSPACE CORPORATION

Notes:
All models should be as simple as possible but no simpler than necessary. (Albert Einstein)
All models are wrong. Some models are useful. (George Box)



Acronyms:

HW: Hardware

SW: Software

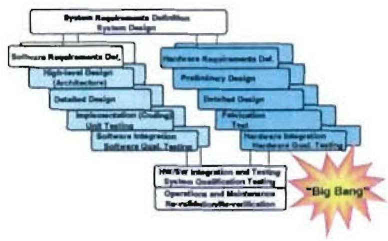
Notes:

Note that the hardware life cycle is also Waterfall (and it is always Waterfall).

The pure Waterfall Model has the following three, key characteristics:

1. All system requirements are defined and allocated to software prior to software design.
2. Software is developed all at once.
3. The software is completely developed and tested prior to systems integration (and the same is true for hardware items). That's how we get to the "Big Bang" creation of the total system.

What is Wrong With This Picture?



"The danger in the sequence is that the project moves from being grand to being grandiose ..."

--- *Harlan Mills*

- **Delayed problem discovery and resolution**
  - Due to the "Big Bang" Integration and System Test approach
- **Design trade-offs can be carried out on system level only**
  - And only at a very early stage
- **Hardware-software units are developed in isolation**
  - Mitigation of hardware and software risks is separated; no opportunity for trade-offs and joint risk resolution
- **All software units are expected to be completed at once**
  - Regardless of size and complexity
- **Assumes an overly simplified, static view of**
  - Requirements
  - Architecture
  - Software entities

Notes:
The model assumes that all concurrently developed software and hardware items, even though they are developed in independent process streams, are completed at the same time, ready for a "Big Bang" integration. It is obvious that this structure does not allow for early validation of requirements, and the resolution of problems at this late stage of the project is more costly.

Despite the shortcomings of the Waterfall Model, why was Winston Royce's 1970 paper [Royce70] that first introduced the model so important? The Waterfall is the "mother of all lifecycle models." It was an attempt to document an existing practice, unlike later lifecycle models that were constructed with the goal of trying to introduce new processes to address various shortcomings of the Waterfall Model.

For fairness' sake it has to be noted that Winston Royce's original Waterfall Model slightly differed from the depiction above on two counts:

(1) Allowed feedback loops between successive stages
(2) Incorporated prototyping into the life cycle, via a "build it twice" step running in parallel with requirements definition and design.

Nevertheless, as the J-STD-016-1995 example above shows it, these steps involving feedback loops in the process have been lost in most descriptions of the model, reinforcing the base pattern as a sequential, once-through Waterfall.

Terminology Interrupt #1

- **Delivery**
  - Delivery as an activity is part of the overall development process
  - Delivery can take place repeatedly, and not only at the end of the development process
  - We can deliver to any Stakeholder
    - It is a common misconception that the recipients of delivery are always the users or customers; we can even deliver to ourselves…
      - Question: Why would we do that?
- **A pattern of confusion …**
  - The following, delivery-related terms are used both as **verbs** and **nouns**:
    - Build
    - Make
    - Release

Terminology Interrupt #1 (Cont.)

- **Build**
  - Noun:
    - A software (system) "build" is defined as a version of the software (system) that delivers a specified subset of the requirements that the completed software (system) will meet
    - To run a simple program, we only have to compile and link it; the process is straightforward, the created build is small
    - A typical, large-scale project involves dozens to even thousands of components and libraries, requiring a more complex build process to create an executable image that can be run on a computer
  - Verb:
    - While the noun refers to a physical object, the verb, as a synonym to the words "construct" or "make", refers to the process of creating that object

Terminology Interrupt #1 (Cont.)

- **Make**
  - Noun:
    - The scripts that are used to automatically create the builds are sometimes called "**make-files**"
  - Verb:
    - The verb **make** is a synonym for the verb **build**, or **construct**
- **Release**
  - Noun:
    - The noun **release** refers to a subset of the end **product**
    - A software (system) release is instantiated through the delivery of a build
  - Verb:
    - The verb **release** refers to the **process** of delivering a subset of the end product

GSAW 2005 – Peter Hantos 15 THE AEROSPACE CORPORATION

Notes:

Releases are further classified as:

- Minor – Major
- Internal – External
- Combinations of the above

Examples:

- A **minor external release** could be a patch correcting a particular defect in a shrink-wrap software package
- A **minor internal release** can simply signify the day's work in a fast-paced development environment (for example, Microsoft,) where producing daily builds is the norm
- A **major external release** can mark the final delivery of the product to the customer
- A **major internal release** could be a version of the software that is available at the first time for integration and test on the target platform

Terminology Interrupt #1 (Cont.)

- **Increment**
  - The difference (delta) between two subsequent releases
  - "Increment" is a conceptual term that in software is instantiated through a "build"
- **Incremental Development**
  - A hardware/software development process that produces partial implementation and then gradually adds preplanned functionality or performance in subsequent increments
- **Incremental Delivery**
  - Incremental Delivery is commonly used as a synonym to Incremental Development in the software engineering practice

Terminology Interrupt #1 (Cont.)

- **Evolution**
  - A process of change in a certain direction
  - A process in which something passes by degrees to a different, more advanced, or more mature stage
  - In common language the word is also used as a synonym for growth or development (e.g., "Child Development")
- **The evolutionary aspect of software development**
  - As early as the mid-1980s the so-called Evolutionary Delivery was introduced as an alternative to the Waterfall Model
    - This strategy promoted frequent delivery of useful results through increments to stakeholders
    - Even though the software is delivered through increments in both cases, Evolutionary Delivery is not the same as Incremental Delivery!
- **Life cycle models vs. LCM patterns**
  - The terms "model" and "pattern" will be used interchangeably
    - "Pattern" reflects the opportunity for repetitive invocation of the applicable model's structure

Terminology Interrupt #1 (Cont.)

- **To avoid confusion, definitions will be used as follows:**
  - The distinction between Evolutionary and Incremental patterns will be based upon whether the requirements can be defined up front:
    - Incremental Pattern:
      - **Requirements are known and understood up-front**
        - Requirements can be planned and allocated to all future increments
    - Evolutionary Pattern:
      - **Not all requirements are known or understood up-front**
        - Requirements can be planned and allocated only to the next increment
- **This use of terminology is consistent with the activity sequencing focus of life cycle modeling**
  - We are concerned about the evolution of requirements, and not the evolution of the objective system or its artifacts

Terminology Interrupt #1 (Cont.)

- **"Once-through" vs. "sequential" vs. "waterfall"**
  - ❖ These terms can be used interchangeably
    - – They are discipline-independent
- **Check your understanding of the new terms:**

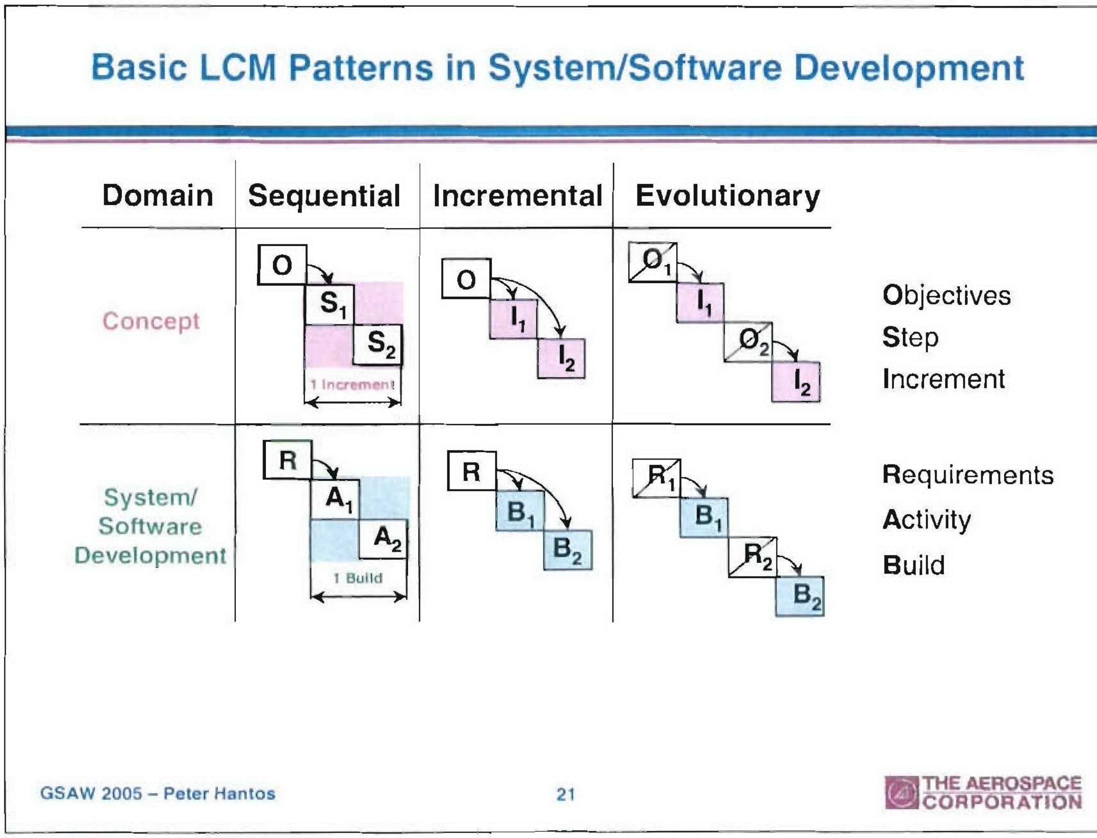
Build	[ ]
Delivery	[ ]
Evolution	[ ]
Evolutionary Delivery	[ ]
Increment	[ ]
Incremental Development	[ ]
Make	[ ]
Model	[ ]
Once-through	[ ]
Pattern	[ ]
Release	[ ]
Sequential	[ ]
Waterfall	[ ]

Overview of Relevant System/Software Development Terms

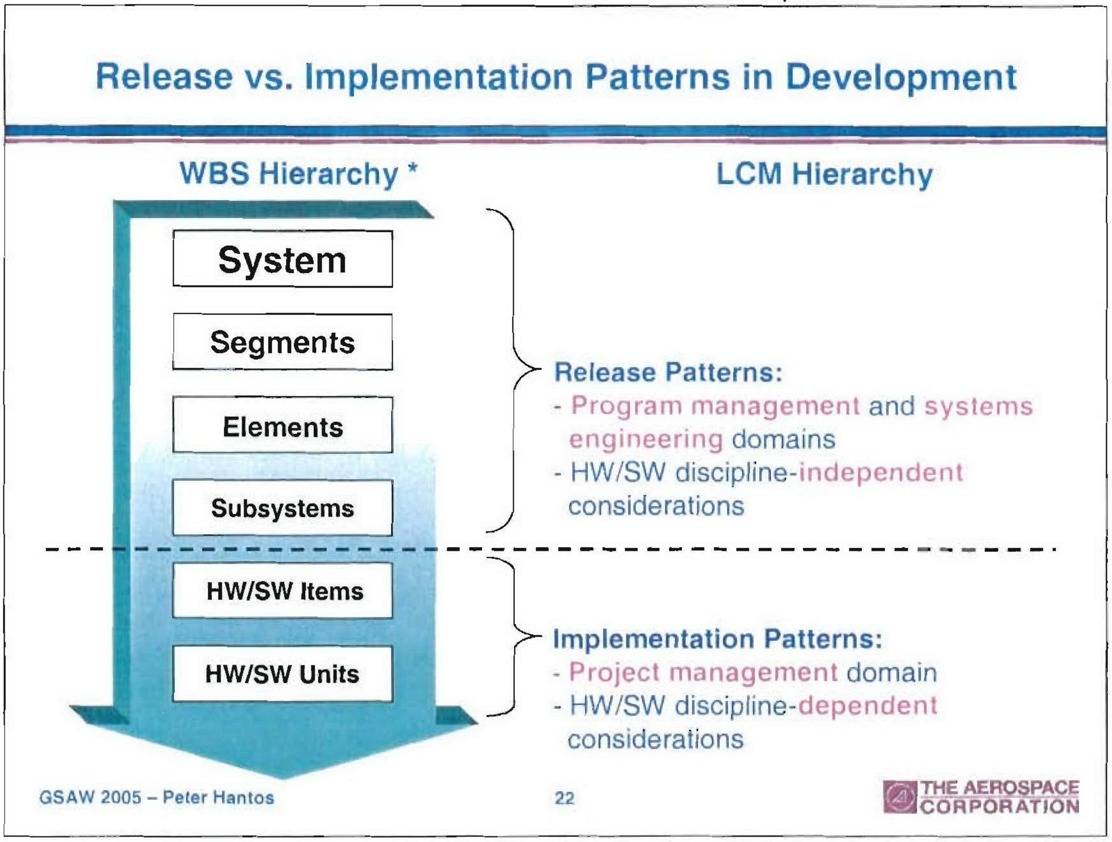
Conceptual Terms	Objectives ... to be accomplished by the process	Increments ... to be completed to achieve part of the objectives	Steps ... to be taken in order to complete one Increment
System/Software Development Terms	Requirements ... given to the engineers to be implemented	Increments ... to be constructed to satisfy some parts of the requirements	Activities ... to be completed in order to create one single Build
		Build ... to be put together to actually deliver an Increment	

Wisdom:
"Divide et impera" ("Divide and rule")
— Roman maxim, 16th Century

Paraphrased Joke:
Q: How do you eat an elephant?
A: One increment at a time!



Notes:
In terms of pattern elements, like Increment and Build that have dual meaning (being used both as verbs or nouns,) the life cycle modeling focus is on the **verb**, i.e., the activity, and not on the created artifacts.



Acronyms:

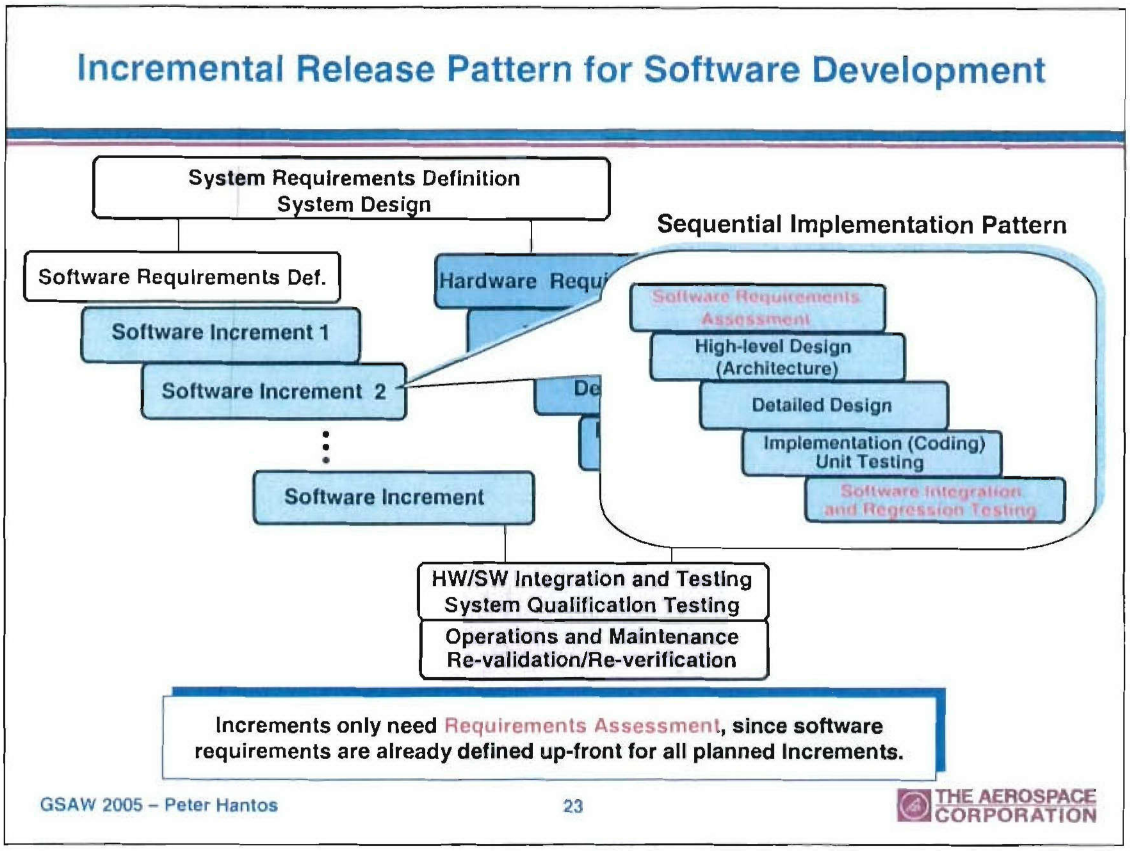
HW: Hardware
LCM: Life Cycle Model
SW: Software
WBS: Work Breakdown Structure

Notes:

* The WBS example shows the space-specific structure; in other software-intensive system development domains, the WBS level-designations are different.

The basic patterns (sequential, incremental, and evolutionary) are applicable in both pattern categories. The later introduced, more complex iterative pattern will be used for implementation patterns only.

For sake of simplicity, the next two charts assume only a two-level WBS hierarchy:
- System
- Software



Acronyms:

HW: Hardware

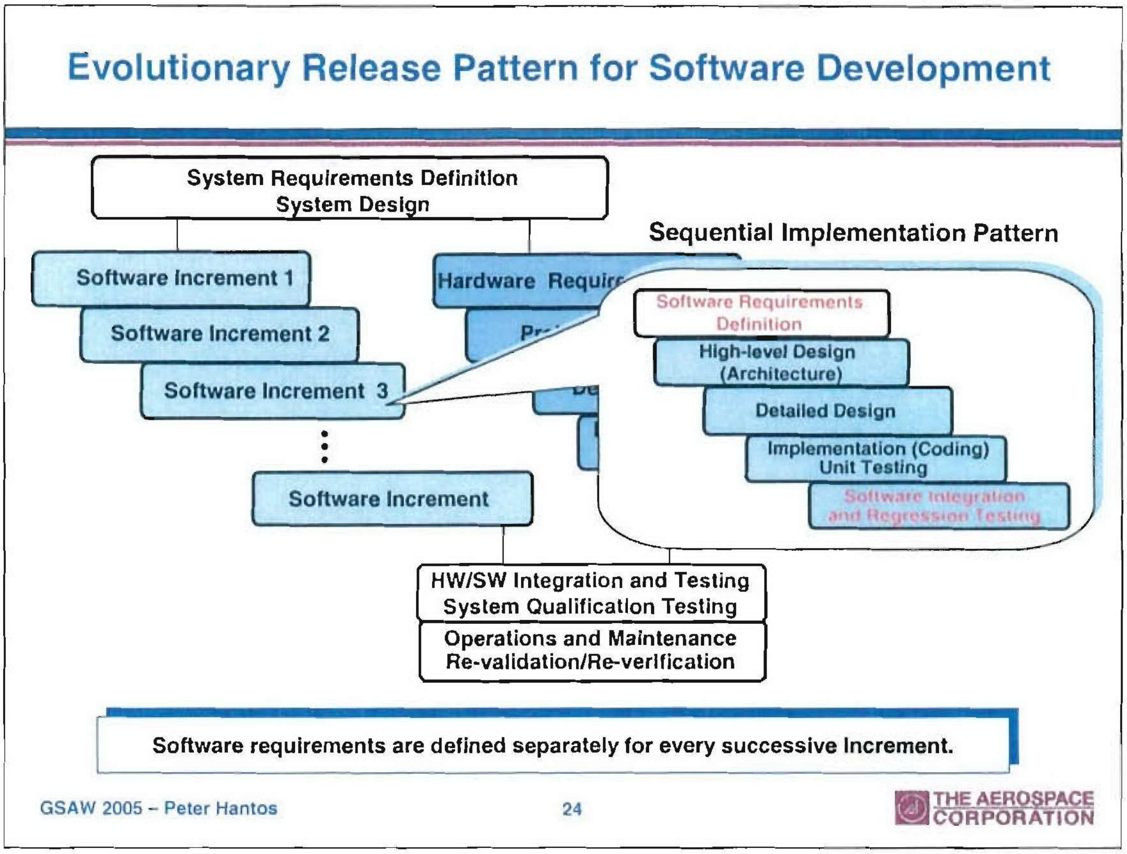
SW: Software

Notes:

- System requirements and design are completed as for Waterfall Model
- System requirements are allocated to software
- Software requirements for all software items are specified and allocated to Increments **up-front**
- Each Increment begins with **Software Requirements Assessment** before development starts, and ends with **Regression Testing** (except, of course, for the first Increment)
- Each Increment delivers a subset of the software's total capability according to the up-front plan
- Each Increment is instantiated through a Build
- The creation of Increments/Builds can overlap in time

The simplified assumption in this example is that the increments are developed, integrated and successively released only in the development environment, on the developers' workstation only. The newly developed hardware is only introduced after the last increment is completed and tested in the development environment; hence the need for HW/SW Integration, Testing, and System Qualification Testing.

A more sophisticated case could be made where successively improving hardware prototypes are becoming available to the software developers. This would be clearly an effective mitigation strategy to discover and handle hardware/software compatibility issues as soon as possible. It is easy to show that the basic LCM patterns are applicable in those, more complex situations as well.



Acronyms:

HW: Hardware

SW: Software

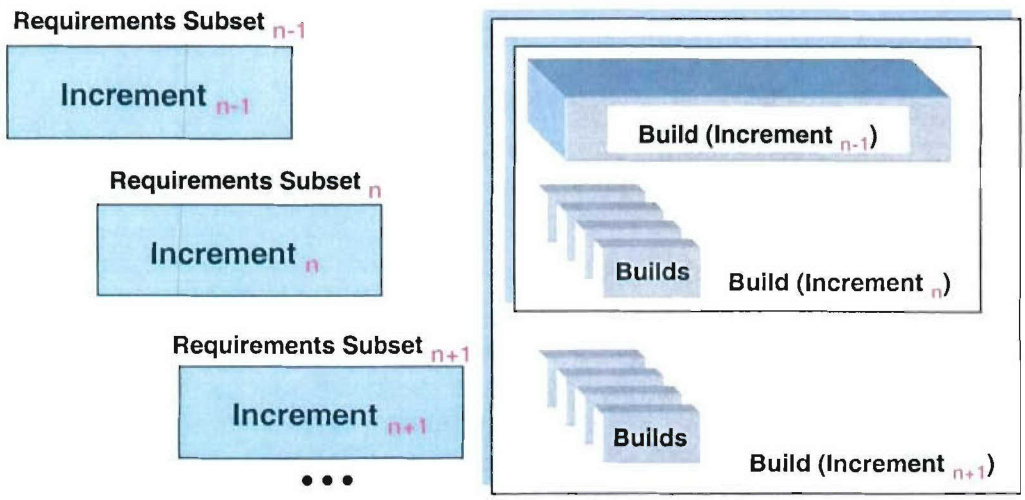
Notes:

- System requirements and design are completed as for Waterfall Model
- System requirements are allocated to software in general, **but are not allocated to Increments**
- Each Increment begins with **Software Requirements Definition** before development starts
- Each Increment ends with **Regression Testing** (except, of course, for the first Increment)
- Each Increment is instantiated through a Build
- The creation of Increments/Builds can overlap in time

Similarly to the Incremental Release Pattern example, the simplified assumption is that the Increments are developed, integrated and successively released only in the development environment, on the developers' workstation only. The newly developed hardware is only introduced after the last Increment is completed and tested in the development environment; hence the need for HW/SW Integration, Testing, and System Qualification Testing.

Again, a more sophisticated case could be made where successively improving hardware prototypes are becoming available to the software developers. It is easy to show here too that the base pattern is applicable in those, more complex situations as well.

Increments and Builds



Note that the objective system (the delivered Builds) grow ("evolve") regardless of the release pattern.

Key Challenges of Build Planning

- **Even if the requirements are believed to be known, i.e. "complete, consistent, correct, clear, feasible, viable, and verifiable," the following challenges exist:**
  - Size/content of Builds
  - Number of concurrently developed Builds
  - Sequencing (development and integration) order of Builds
- **Specific concerns:**
  - How many requirements are truly independent?
  - Which requirements can be logically grouped together?
  - Which requirements are dependent on each other?
  - Are there any expected engineering benefits from a particular implementation order?
  - How many developers are available and what is their skill distribution?
  - How closely does the integration platform match the target platform?

Terminology Interrupt #2

- **Iteration**
  - A procedure in which repetition of a sequence of operations yields results successively closer to a desired result
- **Iterative Development**
  - Involves repetition - iterative, spiral, cyclical are synonyms
  - Iterative development involves learning
    - Create – Review – Change (Improve) on the basis of feedback
  - Iteration is planned revision
    - Work units (scope of iteration) determined by engineering objectives
      - Note that work units of iterations do not necessarily provide additional capability or functionality; the objective might be experimentation or performance enhancement
  - Iteration refers to a period of time
    - The time set aside to revise and improve parts of the system
  - Iteration in development is a risk mitigation mechanism
    - to deal with uniqueness, complexity and technology uncertainties

GSAW 2005 – Peter Hantos 27 THE AEROSPACE CORPORATION

Notes:
The historical basis for iterative development is Walter Shewhart's work from the 1930s, the **P**lan-**D**o-**S**tudy-**A**ct (**PDSA**) cycle for quality improvement.

Terminology Interrupt #2 (Cont.)

- **Now the confusion begins ...**
  - Iterative/Incremental Development (IID) in software engineering
    - Dating back to 1968, the term iterative development implies not only revisiting work, but also evolutionary advancement through increments
- **Our attempt to avoid confusion:**
  - The term "iterative" will be only used to classify LCMs that allow development cycles for revising already completed work
    - Repeating the steps/activities only for the sake of implementing new requirements would not qualify as an iterative pattern
  - This principle will help us to distinguish between the Evolutionary and Iterative LCMs
- **There is still reason for some confusion, because:**
  - The most popular, advanced LCMs that include the iterative pattern, also embrace other base patterns:
    - The Spiral Life Cycle Model is Evolutionary and Iterative
    - The IBM/Rational Unified Process (RUP) is Incremental and Iterative

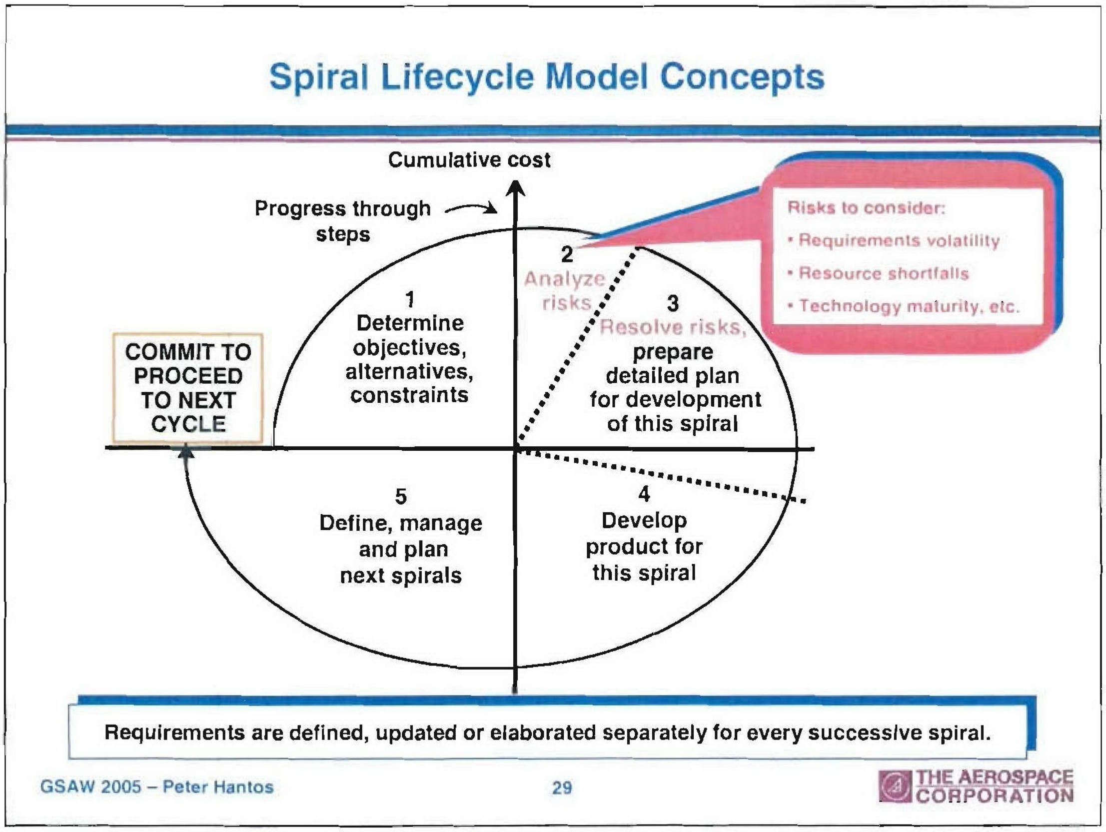
GSAW 2005 – Peter Hantos 28 THE AEROSPACE CORPORATION

Acronyms:
LCM: Life Cycle Model

Notes:
There are numerous life cycle models published in the literature. Most of them are variations or combinations of the basic patterns. For example, **Steve McConnell** in his book, titled **Rapid Development – Taming Wild Software Schedules**, presents the following list of life cycle models:

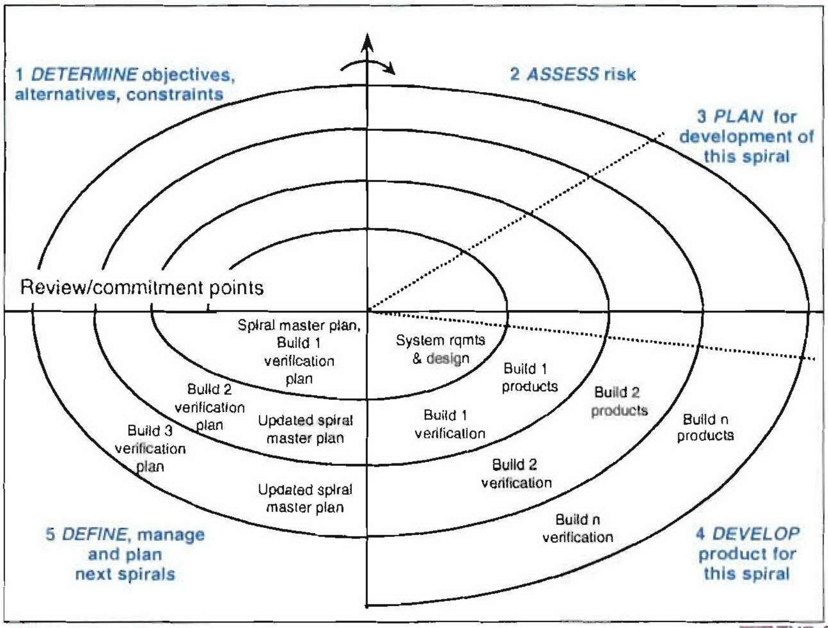
- Pure Waterfall
- Code-and-fix
- Spiral
- Waterfall with Overlapping Phases
- Waterfall with Subprojects
- Waterfall with Risk Reduction Spiral
- Evolutionary Prototyping
- Staged Delivery
- Evolutionary Delivery
- Design-to-schedule
- Design-to-tools
- COTS-oriented

The list is also a good example of the terminology confusion. "Staged Delivery Model" is what we called the "Incremental Release Pattern" and "Evolutionary Delivery" is the same as our "Evolutionary Release Pattern."



Notes:
This is a stylized, simplified version of the spiral that was originally published by Dr. Barry Boehm in 1988 [Boehm88]. The spiral is turning clockwise, representing the direction of progress during development. The key message of the spiral as a metaphor is to show that the development cost is cumulatively growing, even though the same activities are repeated in the appropriate quadrants of the spiral.

Multiple Spirals



Invariant Characteristics* of the Spiral Model

- **Concurrent determination of key artifacts**
  - The process is artifact-driven, not document-driven
- **Each cycle considers critical stakeholder objectives**
  - Stakeholder commitment is obtained on all alternatives
- **Risk-driven determination of level of effort within cycles**
  - Avoids overkill or belated risk resolution
- **Risk-driven determination of degree of detail for artifacts**
  - Avoids overkill or belated risk resolution
- **Managing stakeholder commitments via anchor points**
  - Brings in an architecture-centric management view
- **Emphasis on system and life cycle activities and artifacts**
  - Rather than only software and initial development

GSAW 2005 – Peter Hantos 31 THE AEROSPACE CORPORATION

Notes:

* These invariants were determined during the Y2000 SEI/USC Workshop. Certain difficulties stem from the fact that the spiral diagram, even as it was presented in its original format in Boehm's first article on Spiral Development, either doesn't depict all the key concepts well, or in fact doesn't have some of them at all, because they were invented later.

For example, the title of the original article in 1988 positioned the Spiral as a software development model ("A Spiral Model of Software Development and Enhancement"), and the paper did not mention Anchor Points. On the other hand, the Y2000 workshop gave the following overview definition, dramatically increasing the scope of the model [*Highlights from PH*]:

"The spiral development model is a risk-driven **process model generator**. It is used to guide multi-stakeholder concurrent engineering of **software-intensive systems**. It has two main distinguishing features. One is a cyclic approach for incrementally growing a system's degree of definition and implementation while decreasing its degree of risk. The other is a set of **anchor point milestones** for ensuring stakeholder commitment to feasible and mutually satisfactory system solutions."

The Anchor Point concept first appeared in the literature in 1996 [Boehm96], and it was made, almost simultaneously, part of both the Spiral Model and RUP. More discussion of Anchor Points follows when we introduce RUP.

The IBM/Rational Unified Process (RUP)

- **What is RUP?**
  - A software engineering process
  - A LCM
  - A process product, developed and maintained by IBM/Rational
  - A process framework
  - A collection of selected industry best software engineering practices*
  - A process integrated with a suite of software development tools
- **The underlying development methodology is OO (Object-Oriented)**
  - This is primarily due to historical reasons
    - Fusion with the Objectory process in 1995
    - Fusion with Grady Booch's Object Modeling Technique (OMT) in 1996
- **The underlying LCM is iterative/incremental**
  - RUP documentation refers to it as the dynamic aspect of the process

Notes:
* RUP embraces the following software engineering practices [Krucht99]:
- Develop software iteratively
- Manage requirements
- Use component-based architectures
- Visually model software
- Verify software quality
- Control changes to software

History Lesson: Origins of LCM and Architecture Focus in RUP

- **Iterative Life Cycle Model**
  - Barry Boehm - **Spiral Development [Boehm88]**
    - TRW-SPS (Software Productivity System)/TRW
  - Walker Royce – **Iterative/Incremental Development [Royce90]**
    - CCPDS-R (Command Center Processing and Display System-Replacement) for the US Air Force/TRW
- **Architecture Focus**
  - Phillipe Kruchten – **The 4+1 View Model [Krucht94]**
    - French PBX Systems /Alcatel
    - Canadian Air-Traffic Control System /Hughes
  - Barry Boehm – **Anchor Points [Boehm96]**
    - DARPA (Defense Advanced Research Projects Agency) STARS (Software Technology for Adaptable, Reliable Systems) Program/ Boeing, IBM, and Unisys

Acronyms:

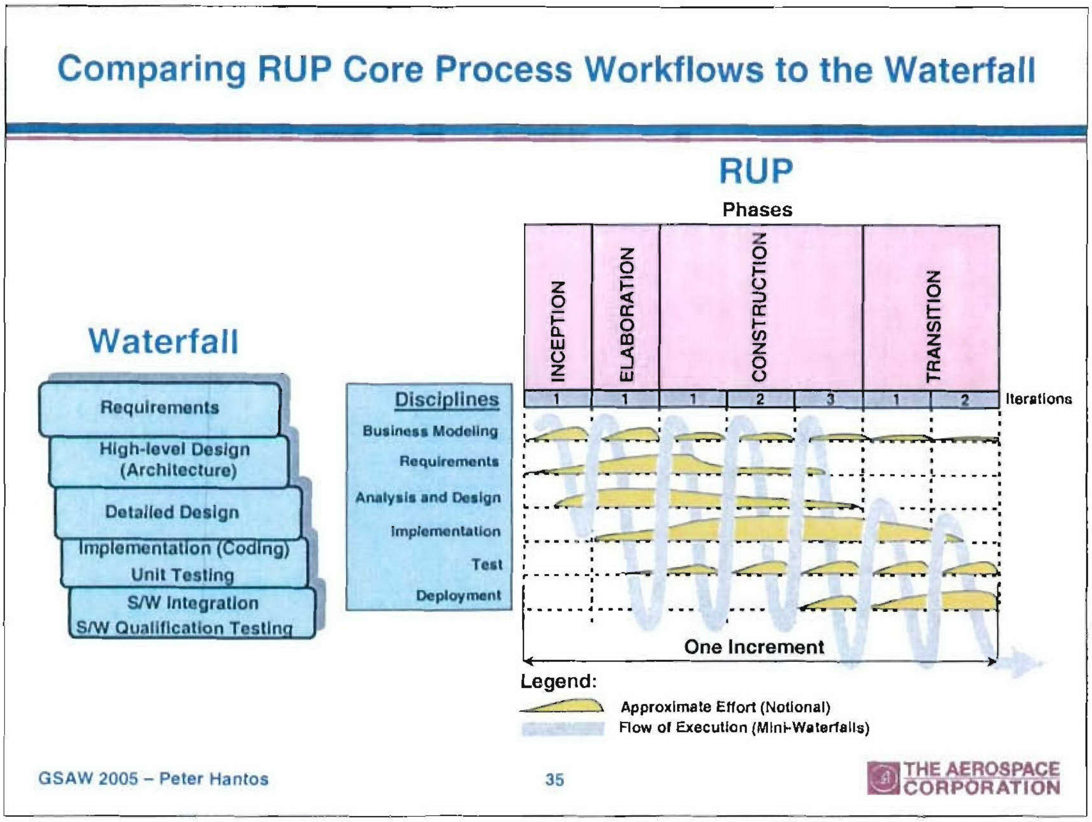
LCM: Life Cycle Model
PBX: Public Branch Exchange
RUP: IBM/Rational Unified Process

Notes:

The learning objective of this, and similar history lessons, is to show why it is often impossible to give a clear-cut definition of many of the concepts and terms we are dealing with. Both the Spiral Model and RUP went through radical metamorphosis over the years, adopting and evolving various aspects of software engineering best practices.

Terminology Interrupt #3

- **Life Cycle Modeling-Oriented Classification of Software/System Engineering Activities**
  - Product-oriented
    - Requirements Definition, Design, Coding, Test, etc.
      - **The RUP terminology refers to them as** Core Process Workflows
      - **All life cycle models show these activities and their relationships**
  - Integral
    - Project Management, Configuration and Change Management, Quality Assurance, etc.
      - **In RUP these are so called** Core Supporting Workflows
      - **Some life cycle models do not show these activities**
        - This is just another, practical aspect of the models' abstract nature



Acronyms:

RUP: IBM/Rational Unified Process
S/W: Software

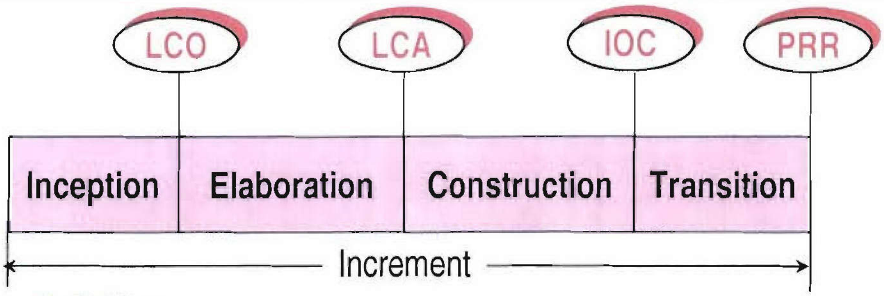
Notes:

The objective of this segment is not to teach the details of RUP. There are numerous resources, books, and last but not least the IBM/Rational website for detailed information. The learning objective is primarily to understand the LCM aspects of the process.

The diagrams only show the fundamental, product-oriented workflows, without explicit reference to the integral workflows. (The Waterfall Models usually don't show integral workflow elements, while the original, published RUP diagram on the IMB/Rational website does.) The RUP diagram is highly conceptual, implying that the iterations inside of the life cycle phases are possibly **mini-Waterfalls,** composed from the activities of the referenced disciplines.

It is also interesting to note that while in the case of the Waterfall Model the author tried to be very specific in describing the steps, the RUP discipline designations are on a very high level. This characteristic of RUP is a further reflection on the fact that RUP is a "unified" model, encompassing details of numerous development methods and processes.

Anchor Points in RUP



- **Definition:**
  - Anchor points are a set of project planning milestones with specific objectives
    - **LCO** (**L**ife **C**ycle **O**bjectives)
    - **LCA** (**L**ife **C**ycle **A**rchitecture)
    - **IOC** (**I**nitial **O**perational **C**apability)
    - **PRR** (**Pr**oduct **R**elease **R**eview)
- **Anchor points bring architecture focus into the life cycle**
- **The need for these anchor points was determined on the basis of studying successful projects**

Anchor Points – Focused Objectives

- **LCO – Life Cycle Objectives**
  - Product-related
    - Definition of operational concept, scope, and top-level requirements
    - Architectural and design options
  - Process-related
    - Life Cycle Plan* defined
- **LCA – Life Cycle Architecture**
  - Product-related
    - Refinement of operational concept, scope, and top-level requirements
    - Resolution of LCO option-explorations
    - Commitment to a feasible architecture and technology solutions
  - Process-related
    - Life Cycle Plan refined

GSAW 2005 – Peter Hantos 37 THE AEROSPACE CORPORATION

Notes:

* The Life Cycle Plan consists of a **global plan** for the whole life cycle, and a **detailed plan** on how the objectives of the next anchor point will be accomplished.

Anchor Points – Focused Objectives (Cont.)

- **IOC – Initial Operational Capability***
  - Product-related
    - Operation and quality is demonstrated in development environment
  - Process-related
    - Readiness for moving to target environment for final implementation, testing and/or integration is demonstrated
- **PRR – Product Release Review**
  - Product-related
    - The work product created in this phase is ready for delivery or higher-level integration
  - Process-related
    - The processes are ready to accomplish the necessary delivery or integration tasks

GSAW 2005 – Peter Hantos 38 THE AEROSPACE CORPORATION

Acronyms:

DoDI: Department of Defense Instructions

NSSAP: National Security Space Acquisition Policy

Notes:

* Unfortunately the acronym IOC is also used in DODI 5000.2 and NSSAP 03-01 with the same meaning, but different content.

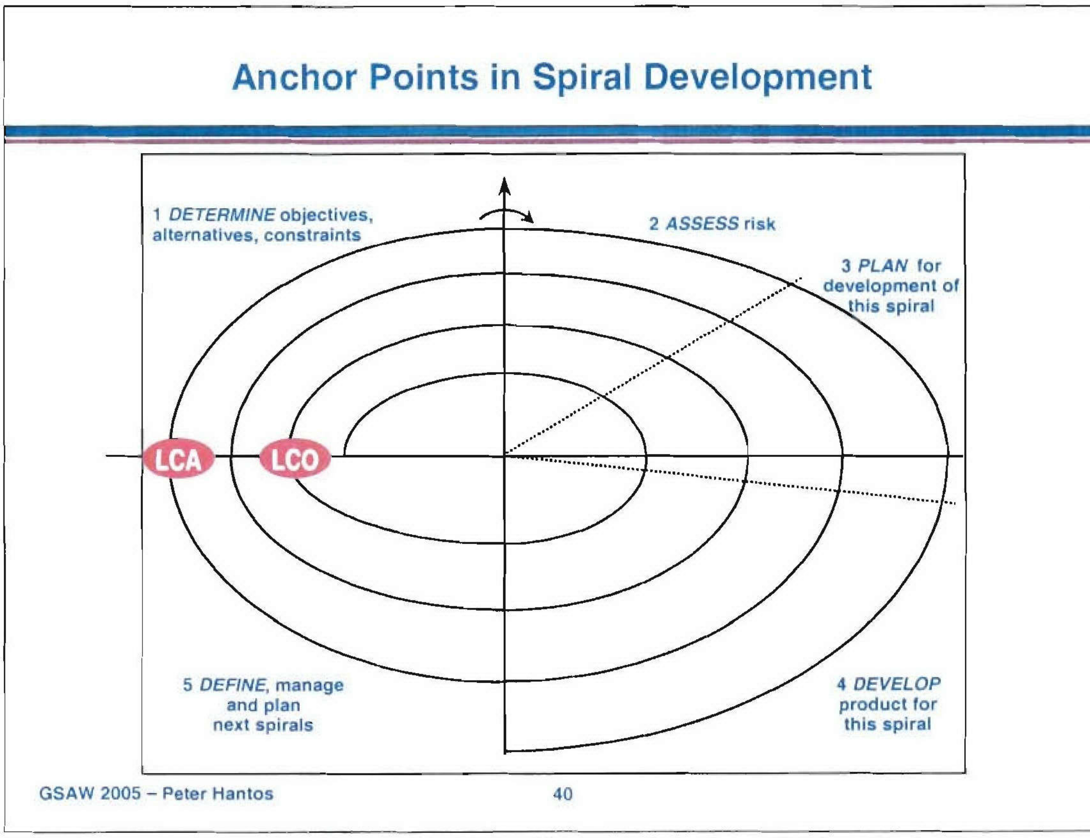
Successful Iterations Are Carefully Planned

- **The objectives of the anchor points are achieved through a sequence of iterations**
  - A coarse-grained, phase plan serves this purpose
- **Iterations themselves need to be planned**
  - A series of fine-grained plans are needed
- **Iteration planning details:**
  - Number of iterations
  - Duration of iterations
  - Content and objectives for iterations
  - Progress tracking
  - Allocating tasks and responsibilities to team members
- **Phase planning and iteration planning are risk-driven***
  - Risk management is not an explicit, core process workflow in RUP, but considered an essential part of iteration planning

Acronyms:
RUP: IBM/Rational Unified Process

Notes:
* The determination of **iteration content** and **iteration sequence** is a risk-driven process. One can chose from three basic iteration planning patterns:
  - Starting with the riskiest, most difficult parts of the task
  - Starting with the easiest parts of the task
  - Letting various user or engineering needs drive the implementation order

Question: What do you think are the pros and cons of the various patterns?



Acronyms:

LCA: Life Cycle Architecture

LCO: Life Cycle Objectives

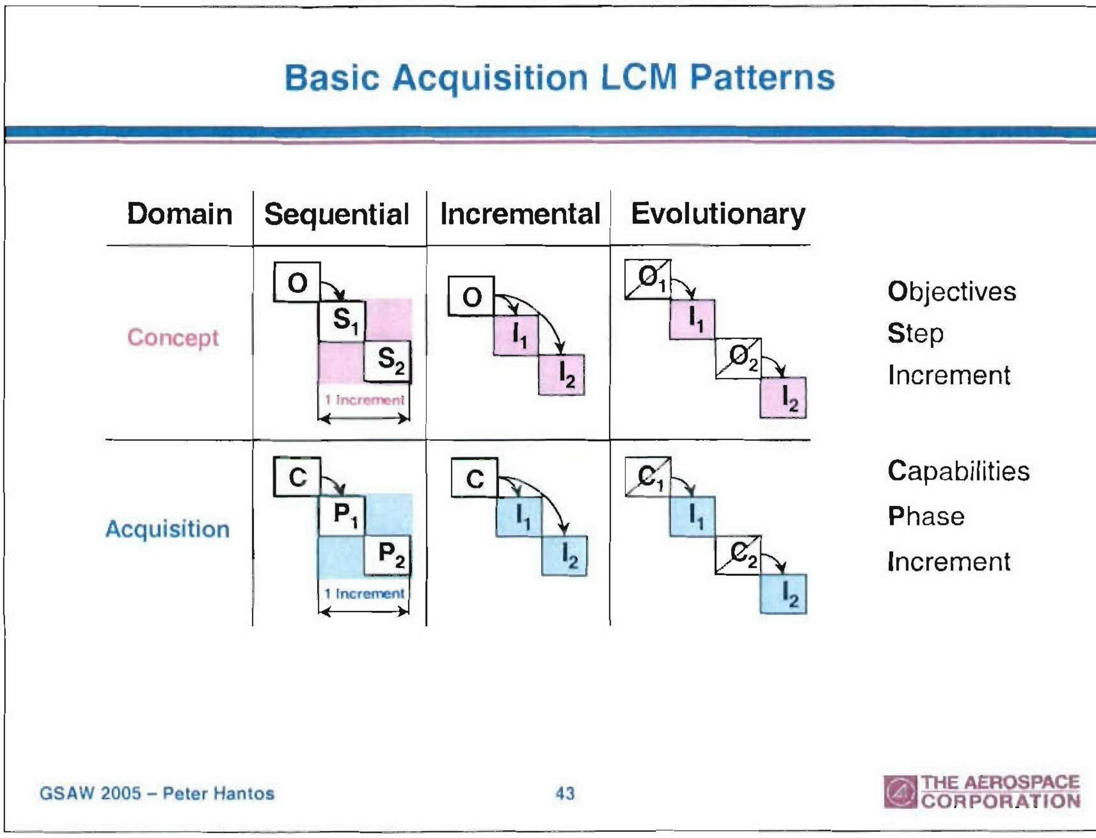
Notes:

Question: Where are the rest of the Anchor Points (IOC, PRR)?

Introduction to Acquisition Life Cycle Models

Overview of Relevant Acquisition Terms

Conceptual Terms	**Objectives** ... to be accomplished by the process	**Increments** ... to be completed to achieve part of the objectives	**Steps** ... to be taken in order to complete one Increment
Acquisition Terms	**Capability** ... to be provided to the government as a result of the process	**Increments** ... to be delivered to provide some parts of the required capabilities	**Phases** ... to be completed while delivering an Acquisition Increment



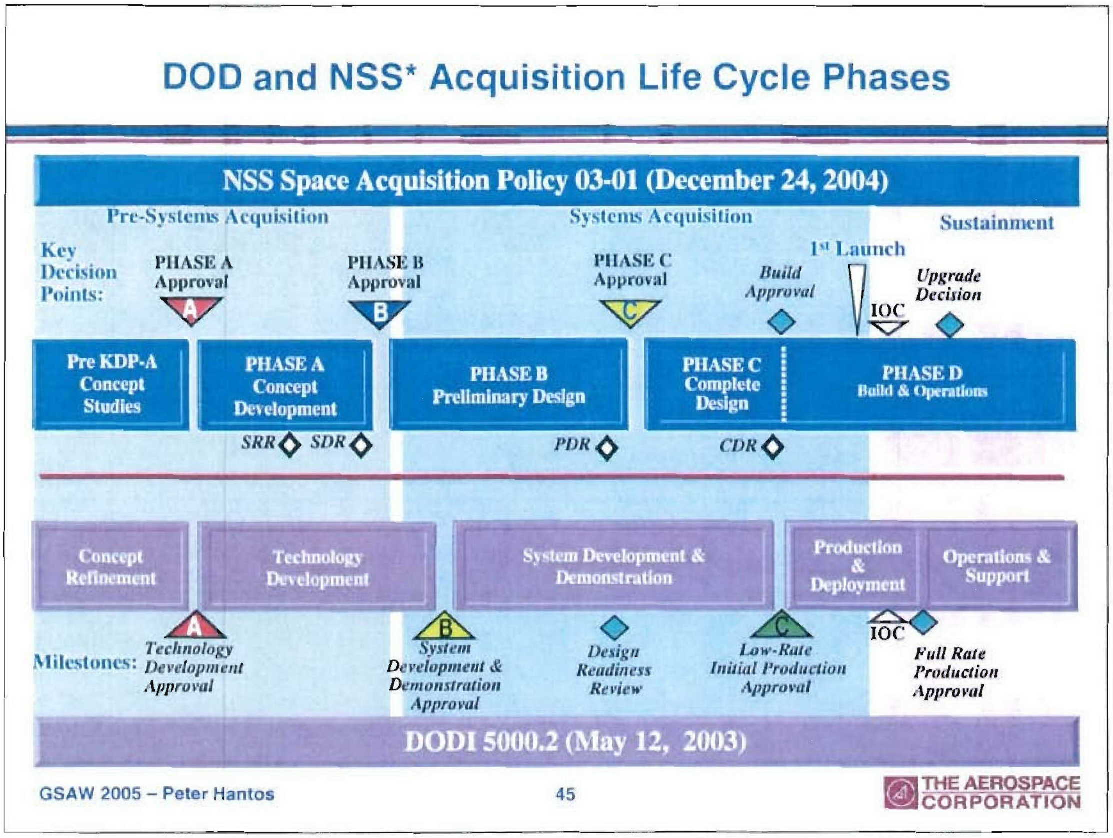
Notes:
This slide demonstrates the common foundation of acquisition and development life cycle patterns.

DOD Space System Types and Associated National Security Space (NSS) Acquisition Models*

- **Satellites, satellite ground systems, and satellite launch vehicle systems**
  - Small Quantity System Model
    - Satellite systems, along with their ground stations and boosters, are usually bought in small quantities
    - Note that these systems are highly software-intensive
- **All kinds of user equipment**
  - Large Quantity Production Focused Model
    - User and data reception terminals
    - These systems are typically bought in quantities of 50 or more

Acronyms:
DOD: Department of Defense

Notes:
* Source: NSS Acquisition Policy 03-01 (December 24, 2004)



Acronyms:

CDR:	Critical Design Review
DOD:	Department of Defense
DODI:	Department of Defense Instructions
IOC:	Initial Operational Capability
NSS:	National Security Space
PDR:	Preliminary Design Review
SDR:	System Design Review
SRR:	System Requirements Review

Notes:

* DODI 5000.2 has a single acquisition life cycle model only. The chart compares the DOD model to the NSS' Small Quantity System Model, showing the first acquisition increment.

Highlights of Evolutionary Acquisition and Spiral Development

- **An approach to deliver capability in increments**
  - It is recognized up-front that new, improved capabilities will be needed in the future
- **Objective is to get mature technology rapidly to the user**
  - Implement/deliver early those aspects of required capabilities that already have their underlying mature technology foundation
    - There is an implied recognition that technology is both the driving and limiting force in weapons system development
- **Why is it a DOD-preferred strategy?**
  - Because it is easy to see that it helps in mitigating anticipated risks, such as:
    - Requirements are volatile
    - Requirements are not well understood
    - New technology is being incorporated
    - Changes of critical technologies might be anticipated
    - Dealing with system complexity and size is a concern

GSAW 2005 – Peter Hantos 46 THE AEROSPACE CORPORATION

Acronyms:

DoD: Department of Defense
NSS: National Security Space

Notes:

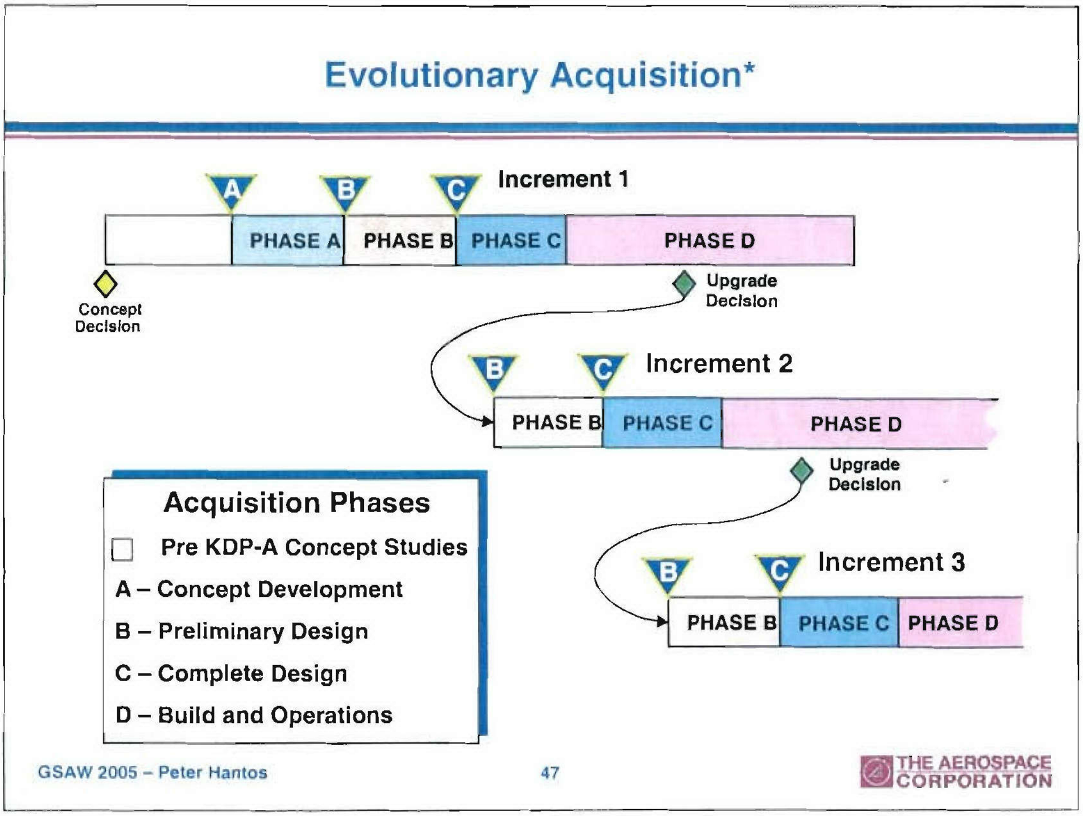
Included is the quote from the actual text of the NSS 03-01 policy, which is basically a repetition of the DOD text:

"AP1.1.3 Evolutionary Acquisition
Within both NSS acquisition models, Evolutionary Acquisition (EA) is the preferred strategy for rapid acquisition of mature technology for the user. EA is defined as an acquisition approach that delivers capability in increments, recognizing up front the need for future capability improvements. This approach requires collaboration among the user, tester, and developer. The two main processes to perform EA are:

a) Spiral Development. In this process, a desired capability is identified, but the end-state requirements are not known at program initiation. Those requirements are refined through demonstration and risk management, there is continuous user feedback, and each increment provides the user the best possible capability. The requirements for future increments depend on feedback from users and technology maturation.

b) Incremental Development. In this process, a desired capability is identified, an end-state requirement is known, and that requirement is met over time by development of several increments, each dependent on available mature technology.

Evolutionary acquisition has been a cornerstone for space system development since the early 1960s. Incremental software and hardware improvements to the ground-based segments of a space system are commonplace. It is also common to perform incremental upgrades on satellites within a space system or constellation."



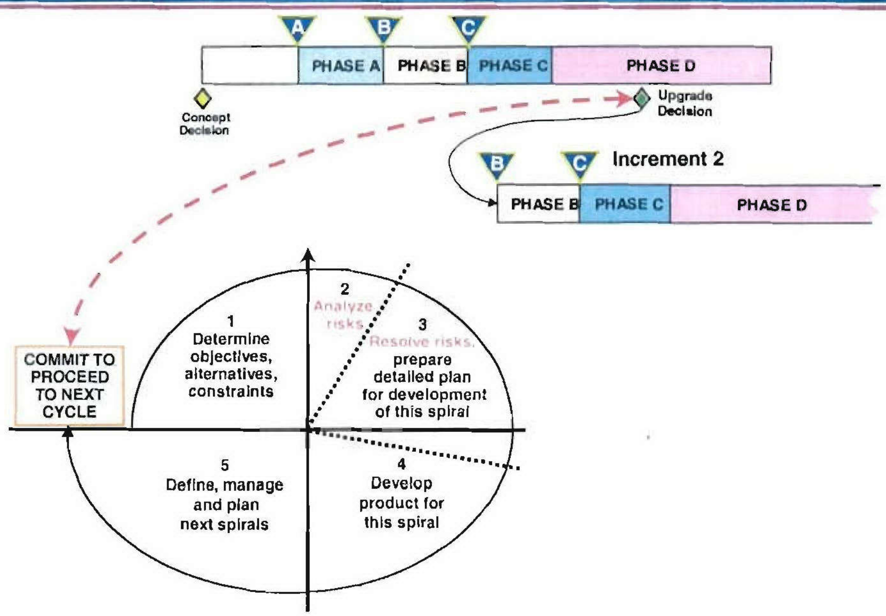
Acronyms:

KDP: Key Decision Point
NSS: National Security Space

Notes:

This slide is based on Figure AP2-3 of the old NSS Acquisition Policy 03-01 (July 28, 2003), and it depicts Evolutionary Acquisition in the context of the Small Quantity System Model. Please note that Figure AP2-3 was omitted from the most recent, December 27, 2004 update of the policy.

But Where Is the Spiral?



Notes:

The spiral is present "in the background" as a process generator. During **Phase D** of the first acquisition increment in the realms of earlier established global capabilities, new requirements are identified. These requirements are matched against the availability of enabling mature technologies, and plans are put together for a new acquisition increment. Please note that the "**Commitment**" step involves all stakeholders of the process, even the U.S. Congress. Consequently, the planning of the next acquisition increment includes the appropriation and budgeting process as well; **Phase B** of the second acquisition increment cannot start unless the necessary funds are available.

Concurrently with the development of the objective system in **Phase D** of the second acquisition increment, threats and other needs are constantly evaluated and matched against available and desired capabilities, and of course technologies, as mentioned earlier. At an appropriate time, when the funding/budgeting outlook is consistent with the needs and technology readiness levels, the cycle can be repeated, and a new, third acquisition incrementmight be initiated.

Who Can Start or End a Spiral for DOD Systems?

- **The acquisition of DOD systems is a result of the coordination of the following three processes:**
  - Joint Capabilities Integration and Development System (JCIDS)
    - Identifies, develops, and validates defense-related capability needs
  - Planning, Programming and Budgeting System (PPBS)
    - Translates the capability needs into budgetary requirements to be presented to Congress for funding
  - Systems Acquisition Process
    - Management and oversight process for the DOD Space Milestone Decision Authority
    - For Space systems, this process is described in NSSAP 03-01

Acronyms:

DOD: Department of Defense

NSSAP: National Security Space Acquisition Policy

The Spiral Confusion Begins ...

- **What is Spiral Acquisition?**
  - It is the same as Evolutionary Acquisition
    - Spiral Acquisition is an unofficial but popular term
- **What is the Win-Win Spiral Model?**
  - The Win-Win Spiral is an enhanced, augmented version of the original Spiral Model
    - It focuses on the recognition that stakeholder dissonance and related political issues can pose a major risk to the project
    - It is based on Barry Boehm's research on the Theory-W decision-making concepts
      - **In this approach we assume that a stakeholder win-win condition exists and a workable compromise can be reached if the right process is chosen**
    - The model includes a new, three-step front-end, facilitating collaborative decision-making among the stakeholders upon entry into a new cycle of the spiral:
      - **Identify next-level stakeholders**
      - **Identify stakeholders' win conditions**
      - **Reconcile win conditions**

The Spiral Confusion Continues ...

- **Can we use the Spiral Model in lower-tier segments/elements/items of a development program?**
  - ❖ The Spiral Model, as an effective iterative approach, can be used at any level of development, regardless of
    - the designation of the overall program
    - the top-level acquisition strategy
  - ❖ More details and an illustration in the Case Study

More Spiral Confusion ...

- **What is a "Spiral Development Program"?**
  - A special, constrained version of the Evolutionary Acquisition strategy for designated, major* defense acquisitions
    - Described in Sec. 803 of 116 STAT. 2604 Public Law 107-314
    - Per NSSAP 03-01:
      - The Space System Program Director/Program Manager (SPD/PM) should describe the program's Evolutionary Acquisition strategy in the program's Acquisition Strategy
      - The Integrated Program Summary (IPS) constitutes the "Spiral Development Plan" for programs using the spiral development process
    - More details on the next slide
  - Caveat: Unfortunately, anybody can call their own programs "Spiral"
    - See the results of the web-search for "Spiral Development Program" references later

GSAW 2005 – Peter Hantos 52 THE AEROSPACE CORPORATION

Acronyms:

DOD:	Department of Defense
IID:	Iterative/Incremental Development
MDA:	Milestone Decision Authority
MDAP:	Major Defense Acquisition Program
NSSAP:	National Security Space Acquisition Policy
USC:	United States Code (in this context; otherwise University of Southern California)
USD(AT&L):	Under Secretary of Defense for Acquisition, Technology, and Logistics

Notes:

* The term is specifically defined by Title USC 2430, and repeated in Paragraph 3 of NSS 03-01, describing the applicability of the policy:

"3.1.1 DoD Space Major Defense Acquisition Programs
A DoD Space MDAP is an acquisition program that is not a highly sensitive classified program (as determined by the Secretary of Defense) designated by the DoD Space MDA or USD(AT&L) as a special interest, or estimated by the DoD Space MDA to require an eventual total expenditure for research, development, test, and evaluation of more than $365 million in fiscal year (FY) 2000 constant dollars; or, for procurement, of more than $2.190 billion in FY 2000 constant dollars."

Notes:

The plan details are clearly structured around the risk management features of the Spiral Model. It is educational to compare this list to the list of guidelines we gave for managing successful Iterative/Incremental Development, because conceptually the same, risk-driven considerations were used. According to those earlier mentioned guidelines, the following plans need to be developed:

- Global Life Cycle Plan for the entire life cycle
- Coarse-grained, phase plan to get to the next anchor point
- Fine-grained iteration plans covering
    - Number of iterations
    - Duration of iterations
    - Content and objectives for iterations
    - Progress tracking
    - Task allocations and responsibilities

In summary, in Spiral Development Programs the key objective is to develop a limited number of satisfactory prototypes; all the constraints are safeguards for preventing the process from going out of control.

The Status on Section 803 Spiral Development Programs

- "... DoD's current draft report states that there are no research and development programs that have been approved as spiral development programs as of September 30, 2003. Section 803 requirements were implemented in DoD Instruction 5000.2, which was effective in May 2003. DoD anticipates that there will be approved spiral development programs to report in 2004."
- Source:
  - *General Accounting Office, **Defense Acquisitions – DoD's Revised Policy Emphasizes Best Practices, but More Controls Are Needed**, Report to the Senate and House Committees on Armed Services, November 2003*

GSAW 2005 – Peter Hantos 54 THE AEROSPACE CORPORATION

Acronyms:
DOD: Department of Defense

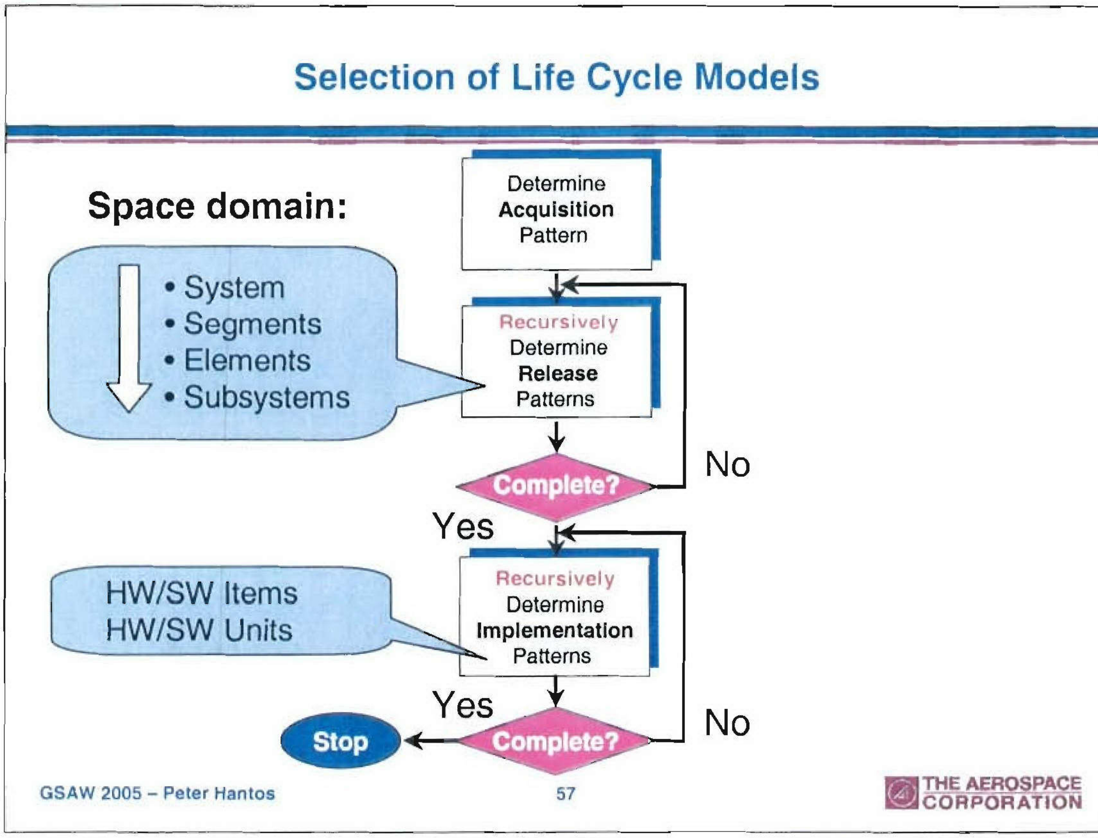
Notes:
In December, 2004 a Google search for "Spiral Development Program" produced about 545 hits. Casual review of those entries showed that people were very liberally using the term, and it was impossible to determine if any of the references were to legitimate, DOD-authorized Section 803 Spiral Development Programs.

Risk-Based Life Cycle Model Selection

LCM Selection is a Risk Mitigation Strategy

- "If you don't actively attack the risks, they will actively attack you." -- *Tom Gilb [Gilb88]*
- Nevertheless, some risks simply cannot be avoided
  - When all risk goes away, so does opportunity …
- LCM selection is the first line of defense for project managers
  - Opportunities and risks of various life cycle models are carefully weighted on the basis of known project characteristics

Acronyms:
LCM: Life Cycle Model



Notes:
The selection order also illustrates the hierarchy of life cycle models.

There are no hard and fast rules to determine when the creation of patterns are "complete." The depth and breadth of planning and the granularity of the releases would depend on the planner or developer's experience and risk awareness, the quality of the development organization, quality of tools, the complexity of the system, and several other factors.

Main Risk Categories

- **Requirements**
  - Requirements volatility
  - Requirements understanding
- **Technology**
  - Hardware and software
  - Technology needed for the objective system
  - Technology used for development and testing
- **Complexity**
  - Dealing with different disciplines (electronics, electro-mechanical hardware, software, materials, optics, etc.)
  - Difficulties with comprehension due to system size
  - Management difficulties due to the large number of people involved
- **Personnel**
  - Quantity (availability)
  - Quality (skills)
- **Politics**
  - Internal - external

GSAW 2005 – Peter Hantos 58 THE AEROSPACE CORPORATION

Notes:

(1) Numerous risk taxonomies exist in the literature, but these risk categories seem to be universal.
(2) The list above applies to both development and acquisition, although in some instances slightly different amplifications or interpretations are needed. For example, requirements on the acquisition level might be referred to as **capabilities**.

Opportunities and Risks of Various LCM Patterns

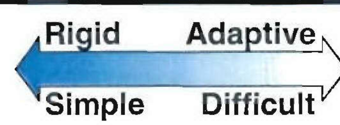
Risk Factor		Implementation							
		Acquisition or Release							
Category	Item	Once-Through		Incremental		Evolutionary		Iterative	
		O	R	O	R	O	R	O	R
Requirements	High requirements volatility is expected due to user feedback		X		X	X		X	
	System is not precedented		X		X	X		X	
	Requirements are not well understood		X		X	X		X	
	User needs some capabilities delivered early		X	X		X		X	
Technology	New technology is being incorporated		X		X	X		X	
	Rapid changes of critical technologies are anticipated		X		X		X	X	
Complexity	Size (SLOC, function points, etc.) is a concern		X	X		X		X	
	High level of inter -dependencies amongst different disciplines		X		X		X	X	
	The system naturally breaks into increments		X	X		X		X	
Personnel	Concerns about responsiveness to funding/staff ing needs		X	X		X		X	
Politics	Concerns about securing funding for a large project		X		X	X		X	
	Difficult stakeholder conflicts are expected		X		X	X		X	

Acronyms:
O: Opportunity
R: Risk
SLOC: Source Lines of Code

The Risk Factor portion is a customized version of the risk list presented on the earlier slide. The objective of the customization is to come up with a table that can facilitate a risk-based determination of all of the project's life cycle models, on every level. Some items, like the skill level of personnel, were left out because, while they were major risk sources, they were equally present for every life cycle model; hence they would not be applicable discriminators for selecting a life cycle model.

Risk-based Acquisition or Release LCM Selection

Risk Factor		Acquisition or Release							
Category	Item	Once-Through		Incremental		Evolutionary		Iterative	
		O	R	O	R	O	R	O	R
Requirements	High requirements volatility is expected due to user feedbac		X		X	X		X	
	System is not precedented		X		X	X		X	
	Requirements are not well understood		X		X	X		X	
	User needs some capabilities delivered early		X	X		X		X	
Technology	New technology is being incorporated		X		X	X		X	
	Rapid changes of critical technologies are anticipated		X		X		X	X	
Complexity	Size (SLOC, function points, etc.) is a concern		X	X		X		X	
	High level of interdependencies amongst different discipline		X		X		X	X	
	The system naturally breaks into increments		X	X		X		X	
Personnel	Concerns about responsiveness to funding/staffing needs		X	X		X		X	
Politics	Concerns about securing funding for a large project		X		X	X		X	
	Difficult stakeholder conflicts are expected		X		X	X		X	



Acronyms:
O: Opportunity
R: Risk
SLOC: Source Lines of Code

Risk-based Implementation-level LCM Selection

Risk Factor		Implementation / Acquisition or Release							
		Once-Through		Incremental		Evolutionary		Iterative	
Category	Item	O	R	O	R	O	R	O	R
Requirements	High requirements volatility is expected due to user feedback		X		X	X		X	
	System is not precedented		X		X	X		X	
	Requirements are not well understood		X		X	X		X	
	User needs some capabilities delivered early		X	X		X		X	
Technology	New technology is being incorporated		X		X	X		X	
	Rapid changes of critical technologies are anticipated		X		X		X	X	
Complexity	Size (SLOC, function points, etc.) is a concern		X	X		X		X	
	High level of inter -dependencies amongst different disciplines		X		X		X	X	
	The system naturally breaks into increments		X	X		X		X	
Personnel	Concerns about responsiveness to funding/staffing needs		X	X		X		X	
Politics	Concerns about securing funding for a large project		X		X	X		X	
	Difficult stakeholder conflicts are expected		X		X	X		X	

Rigid — Adaptive

Simple — Difficult

GSAW 2005 – Peter Hantos 61 THE AEROSPACE CORPORATION

Acronyms:

O: Opportunity

R: Risk

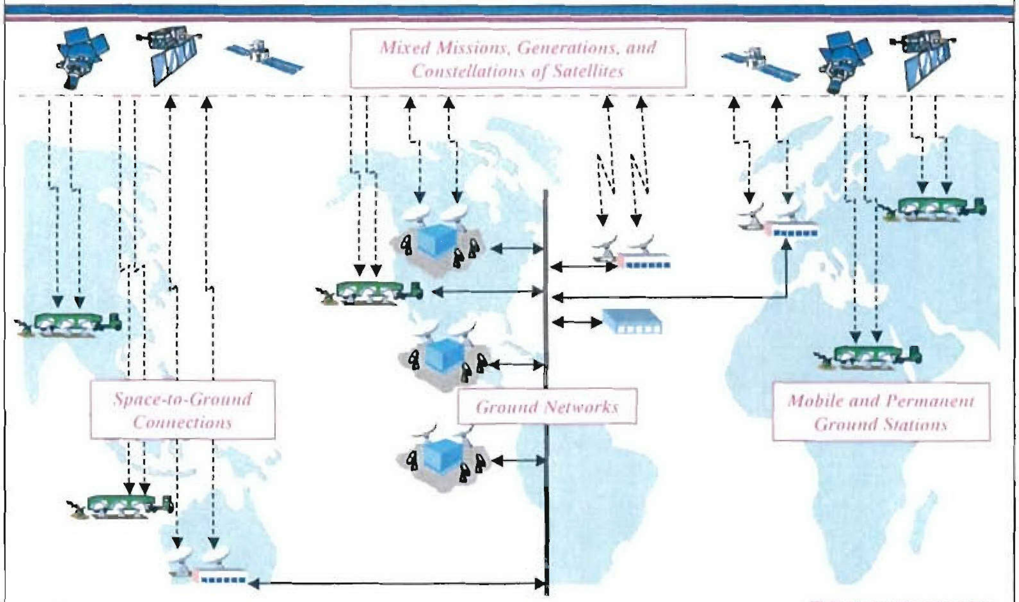
SLOC: Source Lines of Code

This page is intentionally blank

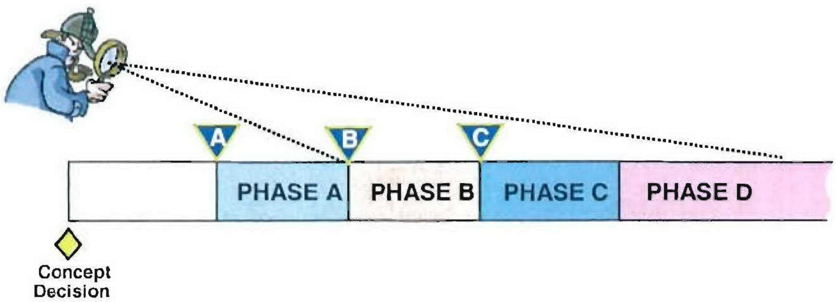
Case Study

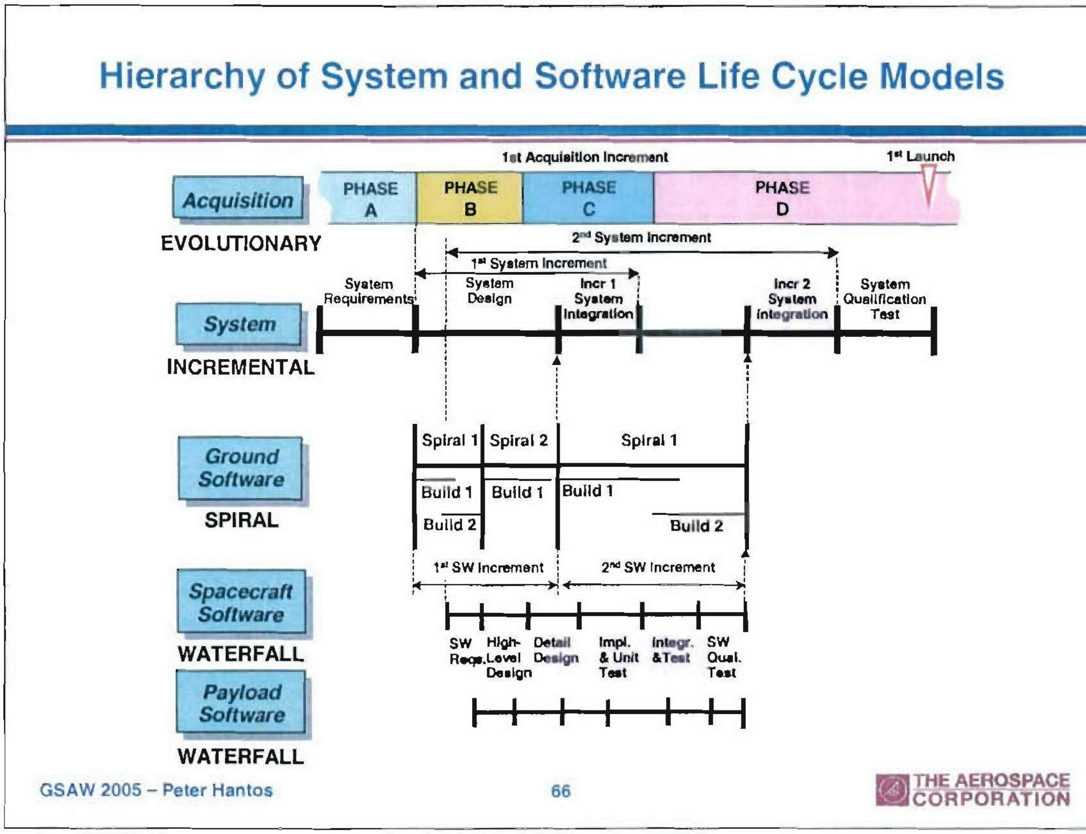
Demonstrating the Hierarchy of Acquisition and Development Life Cycle Models

Complex Space System Example Architecture



First Increment of Evolutionary Acquisition





Notes:
The overall satellite system will be acquired in multiple increments, using Evolutionary Acquisition. In the example, for the sake of simplicity, Hardware Life Cycle Models are omitted.

1st Acquisition Increment:

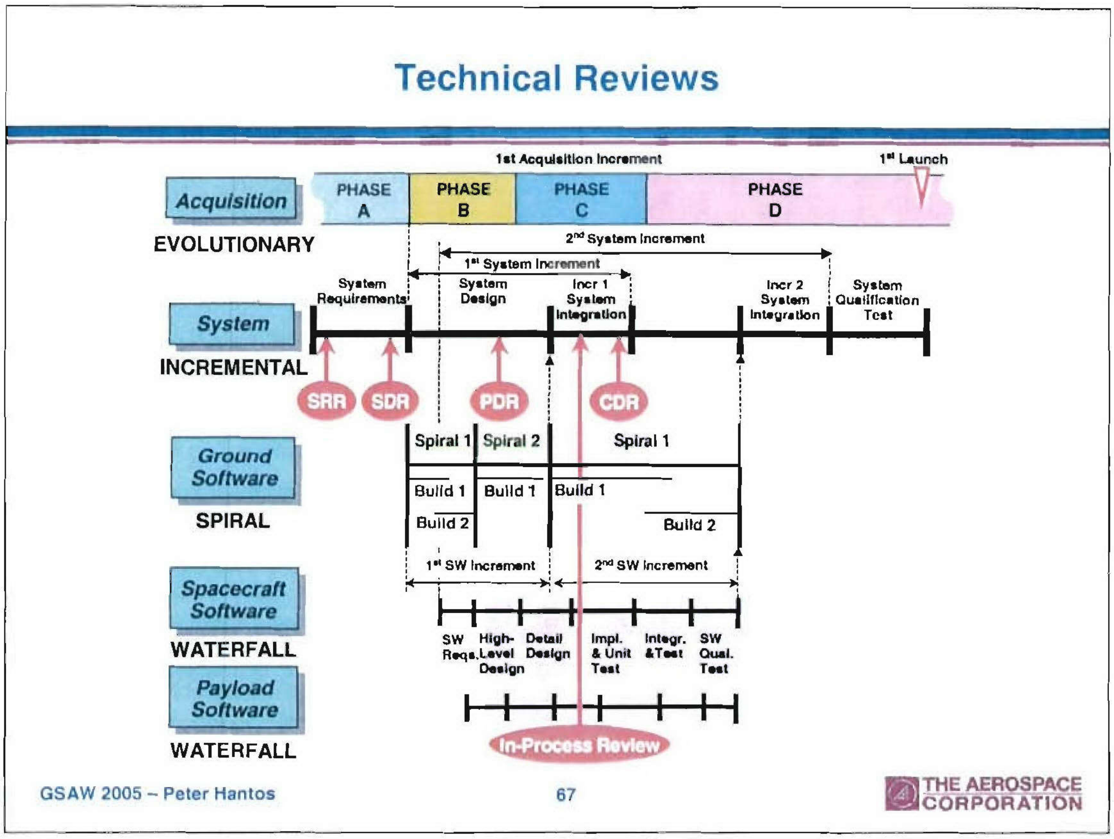
- **Ground System:**
  To be developed in two increments, using Spiral Development.
  - First Software Increment:
    - Development and demonstration of 60% of the necessary new ground system capabilities providing a limited control of the satellites of the existing constellation that will be gradually replaced later.
  - Second Software Increment
    - Development of the remaining 40% of required capabilities.

- **Spacecraft Bus Software:**
  - The plan is to customize a commercially available bus structure. Only one increment is planned, and it is integrated with the fully completed ground software and launched with a few prototype satellites. It is developed using the Waterfall Development Model.

- **Payload Software**:
  - Only limited on-board processing is planned, which can be developed in one single increment using the Waterfall Development Model.

2nd Acquisition Increment (not shown on the diagram):

- This will be a second round in Evolutionary Acquisition; hence the details are not known yet. Some more requirements might be specified for the ground system on the basis of the experience gained during the launch and operation of the prototype satellites. Further satellite payload capability requirements might be determined and a generation of new satellites might be launched.

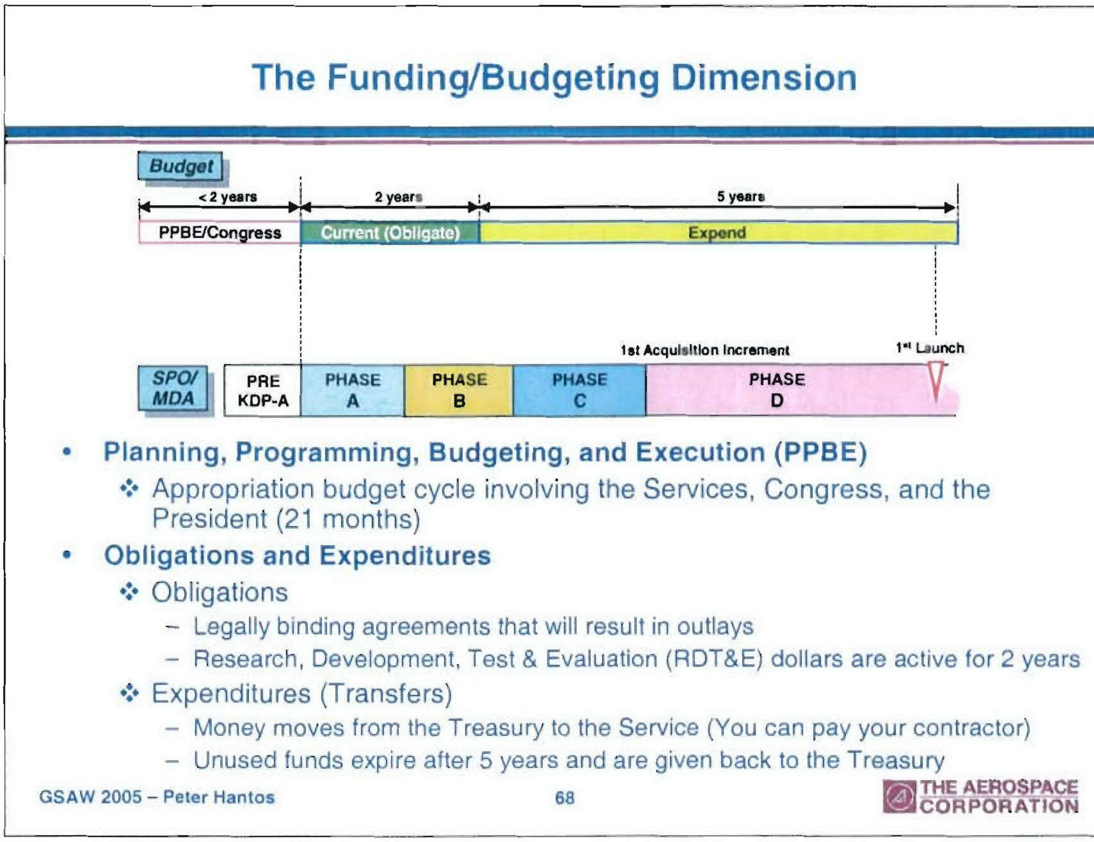


Acronyms:

CDR: Critical Design Review
PDR: Preliminary Design Review
SDR: System Design Review
SRR: System Requirements Review
SW: Software

Notes:

The diagram is meant to illustrate that the naming of PDR and CDR is kind of a misnomer, since at that late stage of the acquisition, software development in all segments progressed way beyond design, even in those categories where the Waterfall life cycle model was chosen. A more appropriate name would be **system-wide In-Process Review**, acknowledging that various artifacts of the different segments would be in different states. The naming and perceived content of those reviews is an unfortunate holdover from the old, **MIL-STD-1521B** conventions and the time when Waterfall was the DOD's standard (and only) development and acquisition life cycle model.



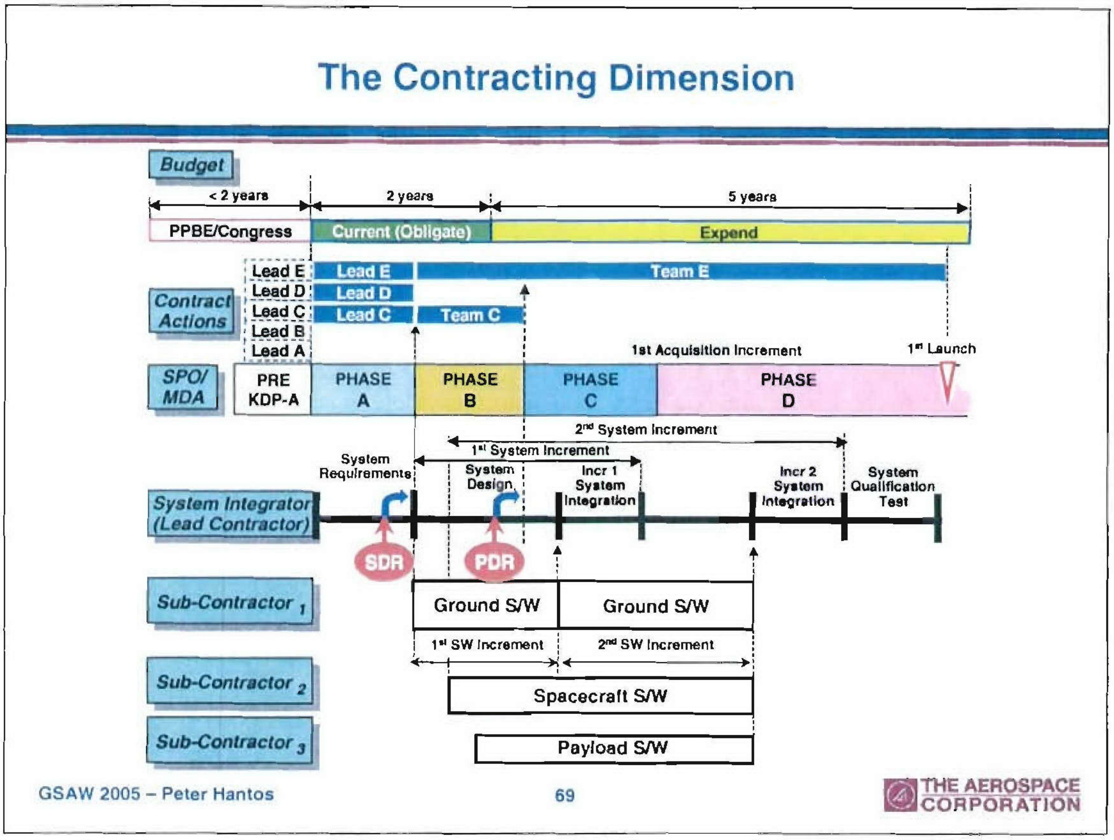
Acronyms:

MDA:	Milestone Decision Authority
KDP:	Key Decision Point
SPO:	Systems Program Office

Notes:

The actual appropriation cycle is even more complicated than what the chart depicts. Congress gives you money for one year's worth of activity. PPBE is repeated every year, and the appropriated funds, even though they belong to the same program, are in different states depending on when they were approved. Congress also monitors the spending rate, and might remove funds from programs that were lagging behind to pay other, urgent, out-of-cycle, mission-critical items. For example, unobligated funds from prior years might be used for an ongoing operation like Iraq or Afghanistan.

PPBE is too complex to explain in one slide and in such a short time. The objective of this and the following few slides is only to show that acquisition planning is a very constrained process, and life cycle models can help to identify and manage the dependencies. For more details, please see [**DODP03**].



Notes:

The earlier life cycle model only showed the technical relationships between the system segments. In reality, a system of this size is developed by a group of contractors, and the contractor involvement adds a new dimension to the budgeting problems during acquisition strategy development.

For the Case Study, the following Acquisition Strategy was chosen:

(1) During the Pre-KDP-A period, five contractors provided concept studies. The Systems Program Office (SPO) has to evaluate the capabilities of all five. This work does not have major budgetary consequences yet; in fact, in some cases the contractors use their own money to bid a contract. The competing contractors are marked as "Leads," because in case of winning the contract, they will act as lead contractors and will engage other contractors, as well, to complete the job.

(2) The result of the KDP-A review is a "down-select" of contractors; only three of them are invited to continue.

(3) Phase A starts with a formal Contract Action, and the three Lead Contractors begin working simultaneously on the requirements and the design of the system, and they engage appropriate subcontractors. The government is contracting only with the Lead Contractors, and the SPO's insight into the financial aspects of the subcontracts is somewhat limited.

(4) At the KDP-B decision point, supported by the System Design Review (SDR) Technical Milestone Review, on the basis of the contractors' performance, only Team C and Team E are allowed to continue the work.

(5) At the KDP-C decision point, supported by the Preliminary Design Review (PDR) Technical Milestone Review, only Team E receives the final approval to finish the job and take the system to its first launch.

Overlapping the contractors is an effective risk mitigation strategy, but very costly. The example demonstrates that all these considerations have to be made very early to ensure approval and funding.

This page is intentionally blank

SPO Clinic

"Ask the Doctor"

Dear Dr. Hantos:

Apparently it is clear that we have to do Spiral Development. Nevertheless, my contractor is telling me that it is planning to use RUP instead. Is RUP a satisfactory replacement for the Spiral?

Sincerely,

— Jane D.
Systems Program Office

The Adequacy of RUP as Spiral Replacement

- **RUP is a legitimate and adequate replacement for Spiral in the software domain on the basis of the following factors:**
  - Architecture focus
    - It is integral part of both models
  - Representative LCM pattern
    - Both emphasize a desirable, iterative approach to development
  - Concurrent engineering
    - Artifact-driven, rather than document-driven processes
  - Risk-based planning
    - While it is more visible in the Spiral, in reality it is an integral part of both models

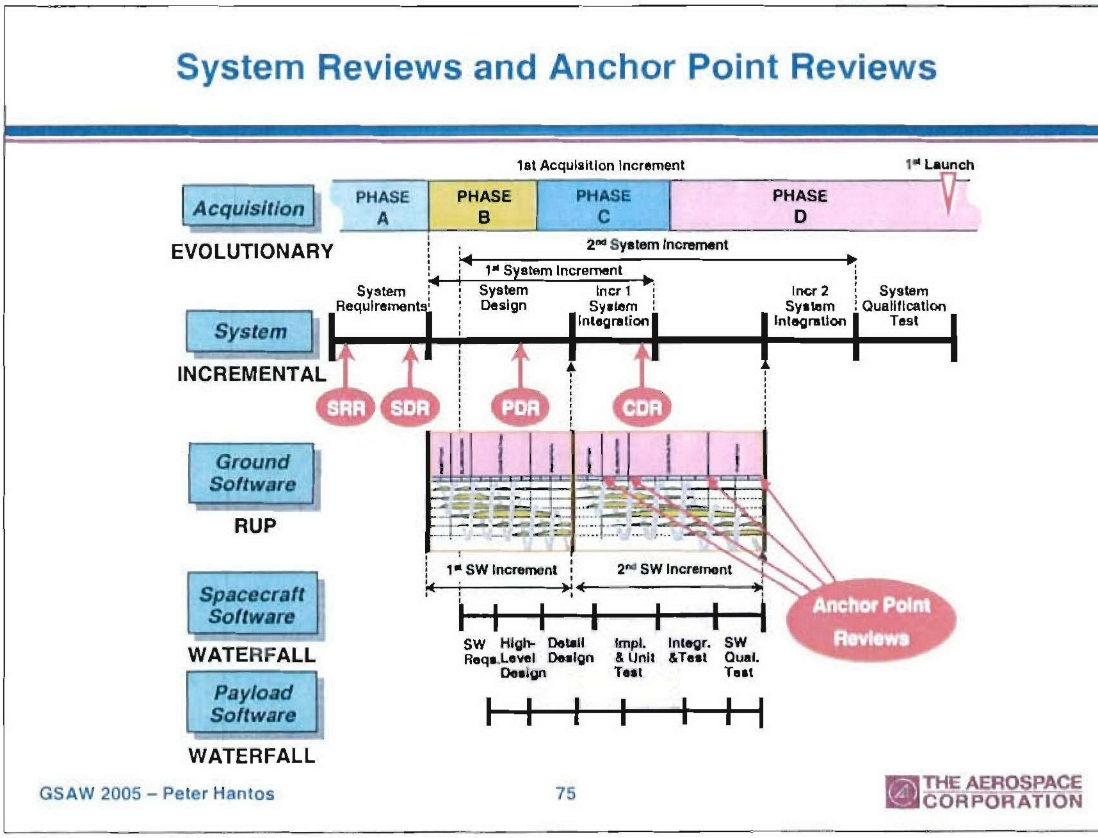
Caveats

- **Differences**
  - The Spiral Model is a more generic process framework
    - It is applicable to both systems and software engineering
  - RUP is highly constrained
    - Difficult to use in a systems engineering context
    - It has strong ties to Object-Oriented Methodologies
    - Details of the process are more formal
- **Some contractors only adopt the tools without the LCM framework**
  - RUP, as a product, includes about 10 different software engineering tools
    - While considerable work was put into the integration of those tools, they are the results of subsequent IBM acquisitions and not conscious, integrated planning
    - Many of them can be used without really implementing an iterative/incremental life cycle model

Acronyms:

LCM: Life Cycle Model

RUP: IBM/Rational Unified Process



<u>Acronyms:</u>

CDR:	Critical Design Review
PDR:	Preliminary Design Review
RUP:	IBM/Rational Unified Process
SDR:	System Design Review
SRR:	System Requirements Review
SW:	Software

This page is intentionally blank

Dear Dr. Hantos:

When I asked about its Spiral Development implementation plans, the contractor said that it, and all of its subcontractors were CMMI Level-5 organizations, so I shouldn't worry. Well, should I?

Sincerely,

— Capt. John D.
US Air Force

Should You Worry?

- **The short answer is YES**
  - Primarily due to the well-known "CMM Math":

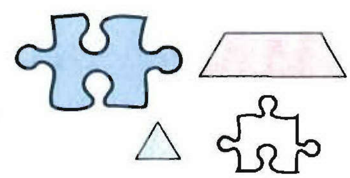
    5 + 5 + ... + 5 = 2
- **But there is more ...**

Acronyms:

CMM: Capability Maturity Model

Explanation of "CMM Math" for Multiple Contractors

- **The prime has its processes and standards**
- **Each subcontractor or teammate has its processes and standards**
- **Each organization's processes and standards may be great for what it does**
- **BUT - they don't necessarily fit together**
  - Within the same company
    - Different product lines
    - Different cultures of divisions or locations
    - Different heritage companies
  - Across companies
- **Specifically, LCMs and Technical Reviews now have to be effectively coordinated and integrated across contractors**



GSAW 2005 – Peter Hantos 79 THE AEROSPACE CORPORATION

Acronyms:
LCM: Life Cycle Model

Issues with Even a Single Contractor

- **Even if the acquisition involves only one CMMI® Level 5 contractor, the CMMI ® rating is no guarantee of effective Spiral Development**
  - It is not very difficult to give only lip-service to LCM-based planning or Spiral Development and achieve a CMMI® Level 5 rating
  - We will discuss some LCM-specific concerns on the next few slides

CMMI® Refresher

- **Level 1 Initial**
  - Process is unpredictable, success depends on heroes
- **Level 2 Managed**
  - Basic project management controls are established on project level
- **Level 3 Defined**
  - Organization infrastructure is established
  - Effective engineering and management processes are in place across all projects of the organization
- **Level 4 Quantitatively Managed**
  - Quantitative objectives and methods for product quality and process performance are used
- **Level 5 Optimizing**
  - Implemented continuous process improvement

GSAW 2005 – Peter Hantos 81 THE AEROSPACE CORPORATION

Acronyms:
CMMI®: Capability Maturity Model Integration

Notes:
Capability Maturity Model, CMM, CMM Integration, CMMI, CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

Reference:
Capability Maturity Model Integration (CMMI, Version 1.1, CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, Supplier Sourcing, Version 1.1, (CMMI-SE/SW/IPPD/SS, V1.1), Staged Representation, SEI-2002-TR-012, Carnegie Mellon University, March 2002.

What Do We Need to Look for at a Minimum?

- **Risk-driven planning**
  - ❖ The Risk Management Process Area must be well integrated with Project Planning, and Project Monitoring and Control Process Areas
    - It is a common problem that there is an active Risk Management Board, but the efforts are disjointed and not coordinated with spiral planning
- **Life cycle model-based project management**
  - ❖ The following process artifacts and evidence of their organization-wide, institutionalized use must exist:
    - Description of approved life cycle models
    - Tailoring guidelines for projects
    - Documentation of defined, tailored processes
- **Quantitative management of the spiral process**
  - ❖ Evidence that process performance is closely monitored
    - Collected measurement data is used to plan successive spirals and improve accuracy of estimation
  - ❖ Caveat: To achieve CMMI Level 4/5 the organization has to only demonstrate quantitative management of selected subprocesses
    - Most organizations only select and deal with defect prevention
    - It is essential to assure that spiral processes are also selected

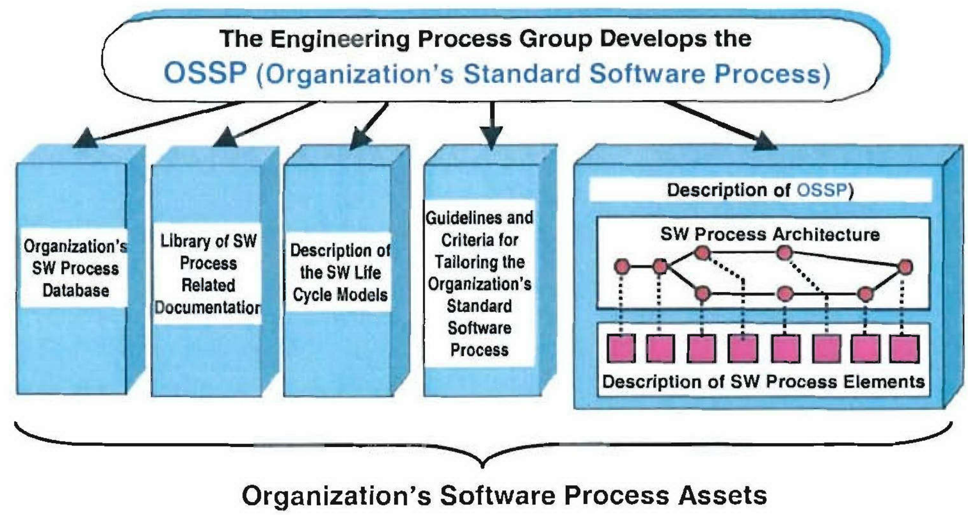
Notes:

The slide only makes a few critical points, and it is not all-inclusive by any means. CMMI Process Areas are heavily dependent on each other, so consequences of being in an Evolutionary Acquisition context and implementing Spiral Development can show up in many more Process Areas (to various extents). For example, the Measurement and Analysis (MA) Process Area would have to outline all the necessary measures that become the basis for a quantitative management of spirals. Similarly, Process and Product Quality Assurance (PPQA) should have the procedures that provide for the continuous monitoring and assessment of spiral planning and execution-related performance. Also, if we would conduct a formal CMMI appraisal, then we would look for evidence showing how the planning of the spiral cycles was improved in the organization via the use of historical performance data. Last but not least, the Decision Analysis and Resolution (DAR) Process Area should be also fully integrated with the spiral planning process.

Defined Processes for LCM-Based Project Management

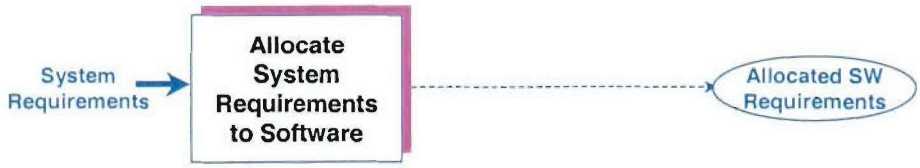
- What is a "Defined" Process?
  - ❖ A process that is tailored from the organization's set of standard processes
    - It is a Generic Goal of the CMMI's Maturity Level 3 that the process is institutionalized as a defined process
- Why use a Defined Process?
  - ❖ Improves project performance
    - Project managers do not have to reinvent the wheel at the inception of new projects and project personnel do not have to learn fundamentally new processes
    - Reduces the amount of work it takes to document project processes
    - Carefully evaluated industry/company best practices are instantly available for project planning
    - Enhances senior management visibility into the projects
    - It serves as the essential foundation for future optimization of the process
  - ❖ Improves project predictability
    - Commonality among projects allows more uniform estimation of performance

Structure of an Organization's Process Asset Library



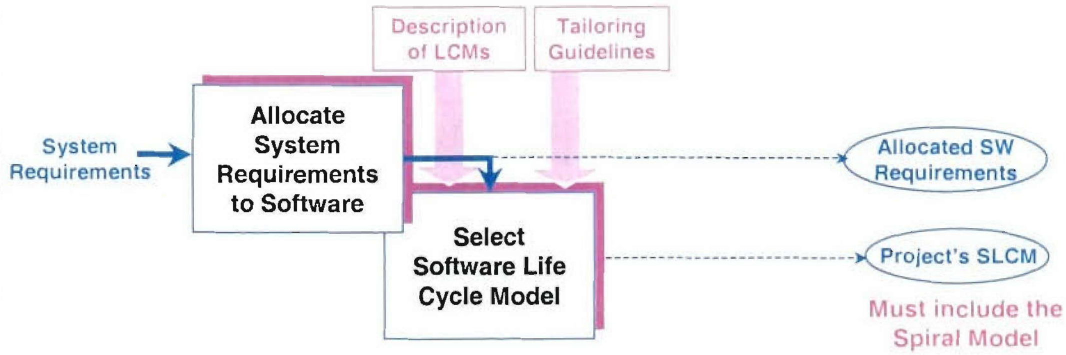
Acronyms:
SW: Software

Steps of Developing a Defined Software Process - 1



Acronyms:
SW: Software

Steps of Developing a Defined Software Process - 2



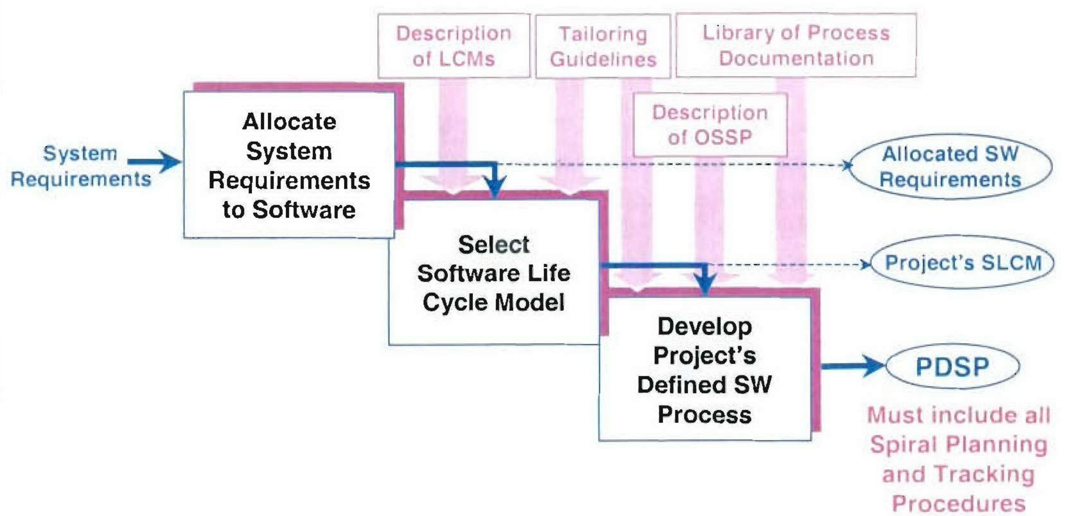
Tailoring guidelines must exist to determine if the use of the Spiral is warranted.

GSAW 2005 – Peter Hantos 86 THE AEROSPACE CORPORATION

Acronyms:

LCM:	Life Cycle Model
SLCM:	Software Life Cycle Model
SW:	Software

Steps of Developing a Defined Software Process - 3



GSAW 2005 – Peter Hantos 87 THE AEROSPACE CORPORATION

Acronyms:

LCM:	Life Cycle Model
OSSP:	Organization's Standard Software Process
PDSP:	Project's Defined Software Process
SLCM:	Software Life Cycle Model
SW:	Software

Could CMMI Level 4/5 Contractors Fail?

- **Past performance is certainly an indicator but not a guarantee of future performance**
  - Despite of the numerous checks-and-balances features of the CMMI, there are no safeguards against institutionalizing bad processes or executing poorly processes that proved to be successful in other settings
  - No insight into and control over staff turnover
- **There is an exposure if the assessed standard set of organizational processes was defined for an earlier and different problem set**
  - Unless your product is identical, optimization will take place at your expense
  - There is no assessment to test the ability to scale-up project scope
- **There is no guarantee that all the processes that the contractor will use on your program were part of the earlier, successful Level 4/5 assessment**
  - During assessment, only selected, representative processes are included
  - It is possible that the contractor organization was recently reorganized

More CMMI Caveats

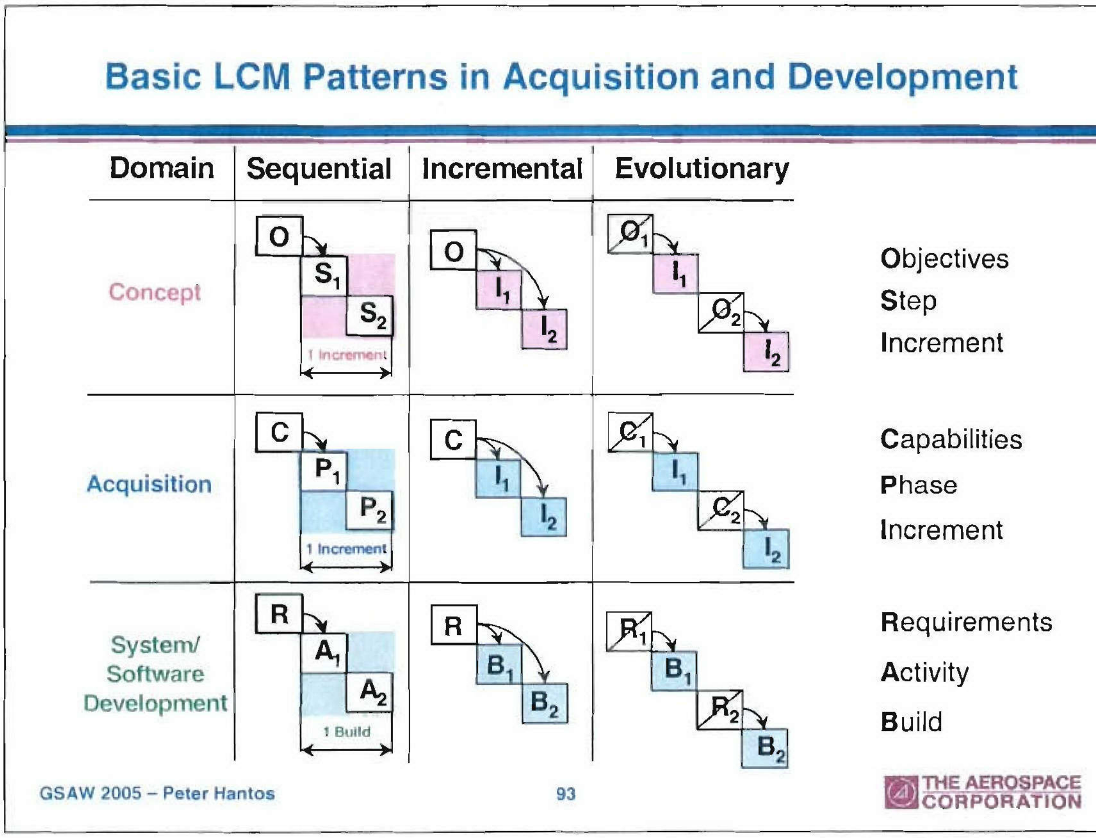
- **Introduction of new technology is always a major challenge**
  - Overly risk-averse risk mitigation strategies can hinder innovation
- **Note on the slides that describe how a Defined Software Process is developed that** System Requirements **drive the definition of the Project's Software Process**
  - Any problems and weaknesses of the requirements set will influence the eventual effectiveness of the Software Development Plan
- **There are no explicit processes in the CMMI to resolve stakeholder conflicts**
  - Even if there was a close monitoring of spiral process performance, spirals can easily go out of control if stakeholders are not cooperating with each other or stonewalling the process

This page is intentionally blank

Summary

Relevant LCM Terms in Acquisition and Development

Conceptual Terms	Objectives ... to be accomplished by the process	Increments ... to be completed to achieve part of the objectives	Steps ... to be taken in order to complete one Increment
Acquisition Terms	Capability ... to be provided to the government as a result of the process	Increments ... to be delivered to provide some parts of the required capabilities	Phases ... to be completed while delivering an Acquisition Increment
System/Software Development Terms	Requirements ... given to the engineers to be implemented	Increments ... to be constructed to satisfy some parts of the requirements Build ... to be put together to actually deliver an Increment	Activities ... to be completed in order to create one single Build



Notes:
This slide demonstrates the common foundation of acquisition and development life cycle patterns.

Conclusions

"Iterative development is not a magic wand that when waved, solves all possible problems and difficulties in software development. Projects are not easier to set up, to plan, or to control just because they are iterative. The project manager will actually have a more challenging task, especially during his or her first iterative project, and most certainly during the early iterations of that project, when risks are high and early failure is possible."

— Philippe Kruchten
IBM/Rational Fellow

Kruchten, P., From Waterfall to Iterative Development – A Challenging Transition for Project Managers, The Rational Edge, 2000

Conclusions (Cont.)

- **Paraphrasing Philippe**

  "Evolutionary Acquisition and Spiral Development are not magic wands that, when waved, solve all possible problems and difficulties in program management or software development. Projects are not easier to set up, to plan, or to control just because they are evolutionary or spiral. The project manager will actually have a more challenging task, especially during his or her first such project."

Conclusions (Cont.)

- **So why should you pursue it after all, if it is so difficult?**

"... A new FBI computer program designed to help agents share information to ward off terrorist attacks may have to be scrapped, the agency has concluded, forcing a further delay in a four-year, half-billion-dollar overhaul of its antiquated computer system.

... An outside computer analyst who has studied the FBI's technology efforts said the agency's problem is that its officials thought they could get it right the first time. "That never happens with anybody," he said."

--- Richard B. Schmitt
Times Staff Writer
*(in a January 15, 2005 LA Times article about the FBI's **Virtual Case File** acquisition efforts)*

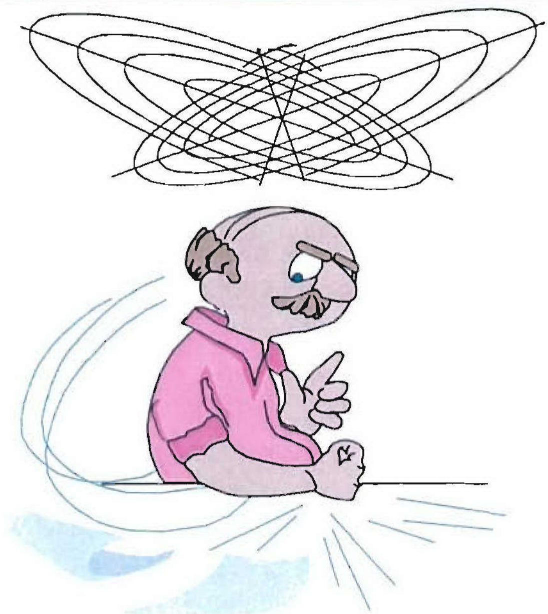
The Final Words

- **Key points to remember:**
  - ❖ Focusing on concepts and patterns instead of administrative details helps navigating around confusing definitions and terminology
  - ❖ Life Cycle Modeling is an effective project management approach
  - ❖ Evolutionary Acquisition and Spiral Development are prudent, risk-driven project management strategies

Test

- **Please answer the following question:**
  The Spiral Model is a
  (a) project management framework
  (b) software life cycle model
  (c) systems engineering process model
  (d) process generator
  (e) ________________________

I Guess You Are Still Confused ...



Homework

- **Let's assume that in the satellite acquisition case study we slightly restructure the objectives by making the completion of the first Ground Software increment the only known deliverable.**
- **Questions:**
  1. Would you still choose an Evolutionary Acquisition strategy?
  2. Evaluate the pro's and con's of this change from the perspectives of
     (a) Congress
     (b) DOD
     (c) Air Force
     (d) SPO
     (e) Contractors
  3. Would the underlying LCM structure change?

Acronyms

CCPDS-R	Command Center Processing and Display System
CDR	Critical Design Review
CMMI	Capability Maturity Model Integration
COTS	Commercial Off The Shelf
CSM	Center for Systems Management
DARPA	Defense Advanced Research Projects Agency
DOD	Department of Defense
EIA	Electronics Industry Association
EVO	Evolutionary Project Management Method
GAO	Government Accountability Office *(formerly General Accounting Office)*
GPS	Global Positioning System
HW	Hardware
IEEE	Institute of Electrical and Electronics Engineers
IID	Iterative/Incremental Development
IOC	Initial Operational Capability
K T/IC	Thousand Transistors per Integrated Circuit
KDP	Key Decision Point
KLOC	Thousand Lines Of Code
LCA	Life Cycle Architecture
LCM	Life Cycle Model
LCO	Life Cycle Objectives
MDA	Milestone Decision Authority
MDAP	Major Defense Acquisition Program
NIST	National Institute of Standards and Technology
NSS	National Security Space
NSSAP	National Security Space Acquisition Policy
O	Opportunity
OSSP	Organization's Standard Software Process
PBX	Public Branch Exchange
PDR	Preliminary Design Review
PDSP	Project's Defined Software Process
PERT	Program Evaluation and Review Technique
PPBE	Planning, Programming, Budgeting, and Execution
PRR	Product Release Review
R	Risk
RDT&E	Research, Development, Test, and Evaluation
RUP	IBM/Rational Unified Process
S/W	Software
SBR	Space Based Radar
SDP	Software Development Plan
SDR	System Design Review
SEI	Software Engineering Institute
SEPG	Software Engineering Process Group
SLCM	Software Life Cycle Model
SLOC	Source Lines of Code
SPC	Software Productivity Consortium
SPO	Systems Program Office
SPS	Software Productivity System
SRR	System Requirements Review
STARS	Software Technology for Adaptable, Reliable Systems
SW	Software
TC	Transformational Communications
USC	United States Code – also, University of Southern California
USD(AT&L)	Under Secretary of Defense for Acquisition, Technology, and Logistics

This page is intentionally blank

Acronyms

CCPDS-R	Command Center Processing and Display System
CDR	Critical Design Review
CMMI	Capability Maturity Model Integration
COTS	Commercial Off The Shelf
CSM	Center for Systems Management
DARPA	Defense Advanced Research Projects Agency
DOD	Department of Defense
EIA	Electronics Industry Association
EVO	Evolutionary Project Management Method
GAO	Government Accountability Office *(formerly General Accounting Office)*
GPS	Global Positioning System
HW	Hardware
IEEE	Institute of Electrical and Electronics Engineers
IID	Iterative/Incremental Development
IOC	Initial Operational Capability
K T/IC	Thousand Transistors per Integrated Circuit
KDP	Key Decision Point
KLOC	Thousand Lines Of Code
LCA	Life Cycle Architecture
LCM	Life Cycle Model
LCO	Life Cycle Objectives
MDA	Milestone Decision Authority
MDAP	Major Defense Acquisition Program
NIST	National Institute of Standards and Technology
NSS	National Security Space
NSSAP	National Security Space Acquisition Policy
O	Opportunity
OSSP	Organization's Standard Software Process
PBX	Public Branch Exchange
PDR	Preliminary Design Review
PDSP	Project's Defined Software Process
PERT	Program Evaluation and Review Technique
PPBE	Planning, Programming, Budgeting, and Execution
PRR	Product Release Review
R	Risk
RDT&E	Research, Development, Test, and Evaluation
RUP	IBM/Rational Unified Process
S/W	Software
SBR	Space Based Radar
SDP	Software Development Plan
SDR	System Design Review
SEI	Software Engineering Institute
SEPG	Software Engineering Process Group
SLCM	Software Life Cycle Model
SLOC	Source Lines of Code
SPC	Software Productivity Consortium
SPO	Systems Program Office
SPS	Software Productivity System
SRR	System Requirements Review
STARS	Software Technology for Adaptable, Reliable Systems
SW	Software
TC	Transformational Communications
USC	United States Code – also, University of Southern California
USD(AT&L)	Under Secretary of Defense for Acquisition, Technology, and Logistics

This page is intentionally blank

Selected Web Resources

- SEI (Software Engineering Institute)
  - http://www.sei.cmu.edu
- SSCI* (Systems and Software Consortium, Inc.)
  - http://www.systemsandsoftware.org
- AT&L (Acquisition, Technology, and Logistics) Knowledge Sharing System
  - http://akss.dau.mil/jsp/default.jsp
- IBM/Rational Unified Process
  - http://www-306.ibm.com/software/awdtools/rup/

Notes:

* **SSCI** used to be called **SPC** (Software Productivity Consortium)

THE AEROSPACE
CORPORATION
2350 E. El Segundo Boulevard
El Segundo, California 90245-4691
U.S.A.