NORTHROP GRUMMAN

Best Practices for Systems Integration

November 2011

Presented by
Pete Houser
Manager for Engineering & Product Excellence
Northrop Grumman Corporation

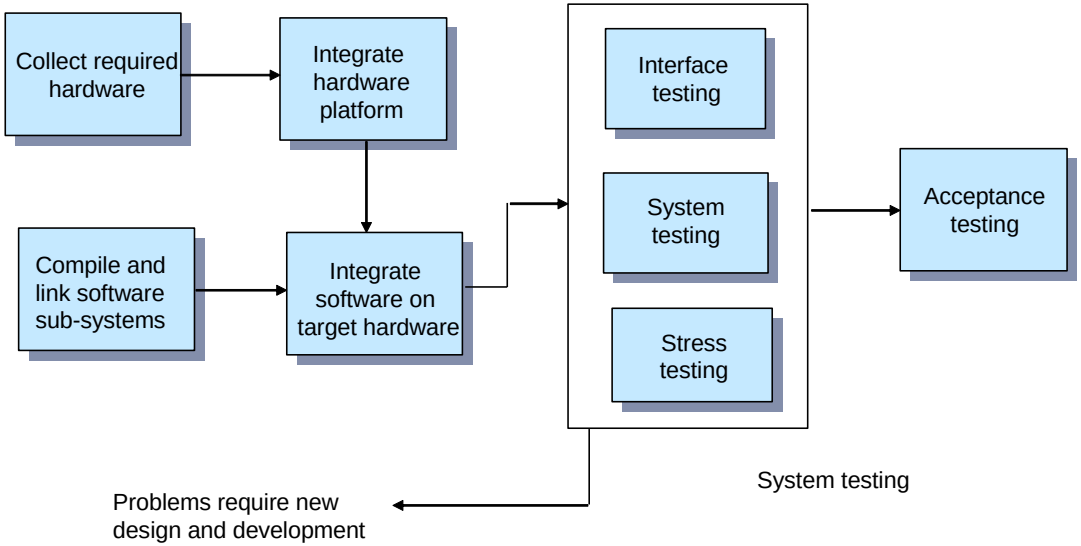
Systems Integration Definition

NORTHROP GRUMMAN

- Systems Integration (SI) is one aspect of the Systems Engineering, Integration, and Test (SEIT) process. SI must be integrated within the overall SEIT structure.
- Systems Integration is the process of:
  - Assembling the constituent parts of a system in a logical, cost-effective way, comprehensively checking system execution (all nominal & exceptional paths), and including a full functional check-out.
- Systems Test is the process of:
  - Verifying that the system meets its requirements, and
  - Validating that the system performs in accordance with the customer/user expectations
- Across Dod Programs, systems integration experiences have been erratic (at best). In many cases:
  - Programs do not define dedicated Integration Engineers.
  - Immature system is passed to the Test organization for concurrent Integration and Test.
  - Program budget is exceeded because Integration was not separately estimated.

Systems Integration Issues

- **When executed as a distinct process, Systems Integration has historically followed the "big bang" model shown below.**
  - All of the components arrive more-or-less simultaneously (usually late) in the Integration Lab.
  - The Integration engineers arrive more-or-less simultaneously then try to make each piece work.
  - Test gets the system when everything is working, or delivery minus - 1 day, whichever comes first.



- **Many symptoms illustrate that this process is flawed:**
  - Integration starts late because of late components, resulting in a flawed and immature delivery.
  - Integration is expected to occur instantaneously, feeding an abbreviated and incomplete Test process, in order to recover schedule and meet the delivery deadline.
  - Focus is on components rather than system capabilities;
    the Integration team rarely understands the overall system concepts.
  - Integration starts when the hardware and software are ready,
    so it uses the delivered hardware and software for all activities.

Recommended Process Changes

NORTHROP GRUMMAN

1. Adopt a Continuous Integration model rather than a Big Bang Integration model.
   Establish an Integration rhythm that is essentially independent of the development team.

2. Manage system integration and system test based upon subsystems that can be end-to-end tested against system level requirements; manage system design & development based upon components that can be independently developed and checked.

3. Create a Systems Integration team of Responsible Engineers that knows the entire system and follows the program from Requirements Definition through Acceptance Testing and Operations. Design and Test engineers provide required support to REs during integration.

4. Create a System Architecture Skeleton (SAS) very early in the program and use it as the framework for Subsystem Integration as components are added incrementally

5. Define a Configuration Management process such that the System Integration and Configuration Management Teams build and control the hardware & software configurations.

6. Develop component and subsystem specifications to the extent that they are needed in order to define component checkout & subsystem verification procedures.

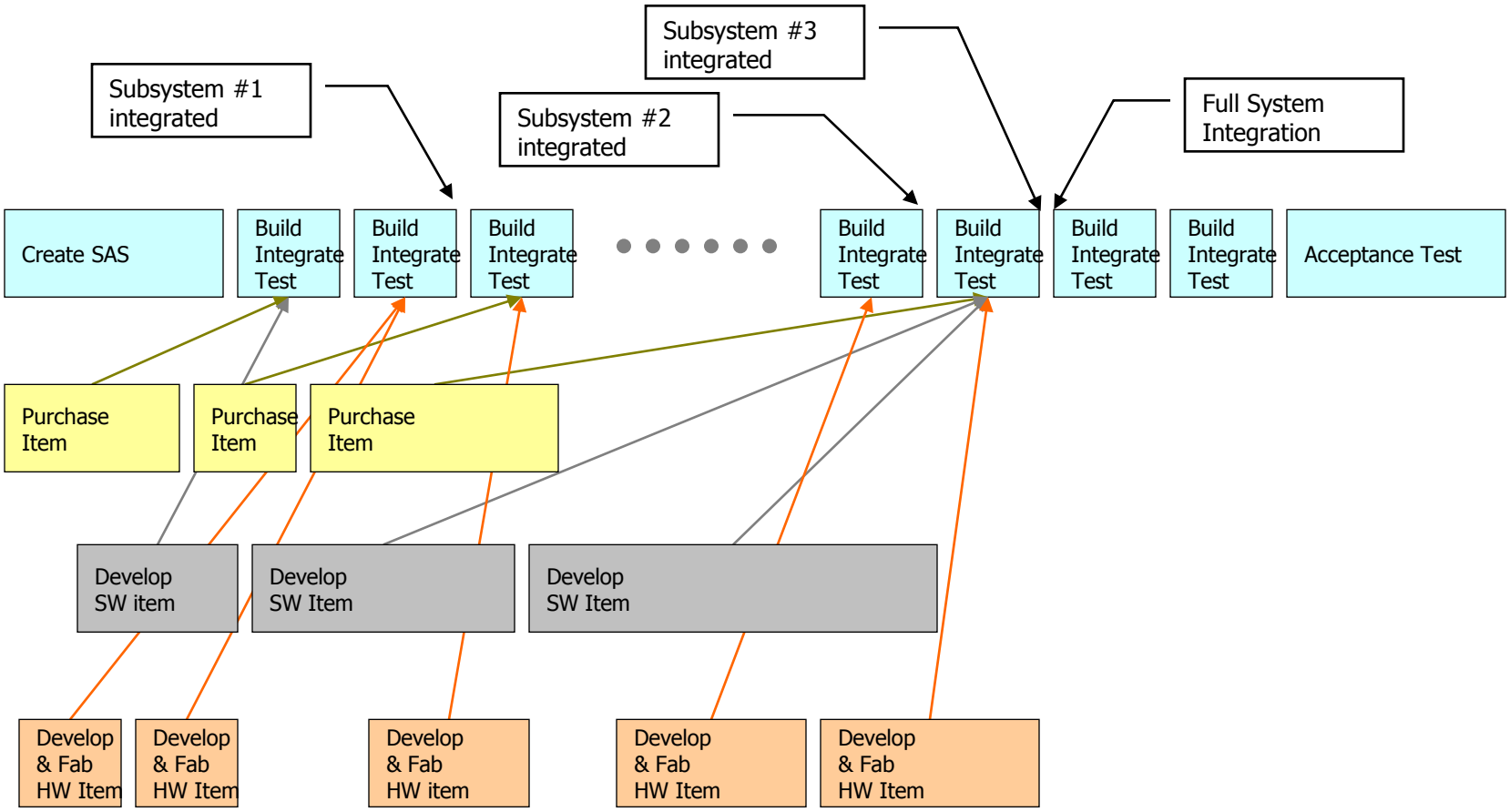
7. Perform component-level checkout to satisfy Integration entry criteria.

8. Track integration progress based upon completing subsystems that have been verified end-to-end against system-level requirements.

9. Continuously perform regression testing; create internal and/or external automated test tools that greatly reduce the emphasis on man-in-the-loop testing.

10. Augment requirements-driven testing with stress testing and long mission threads testing in order to promote a robust system.

1: Continuous Integration Model



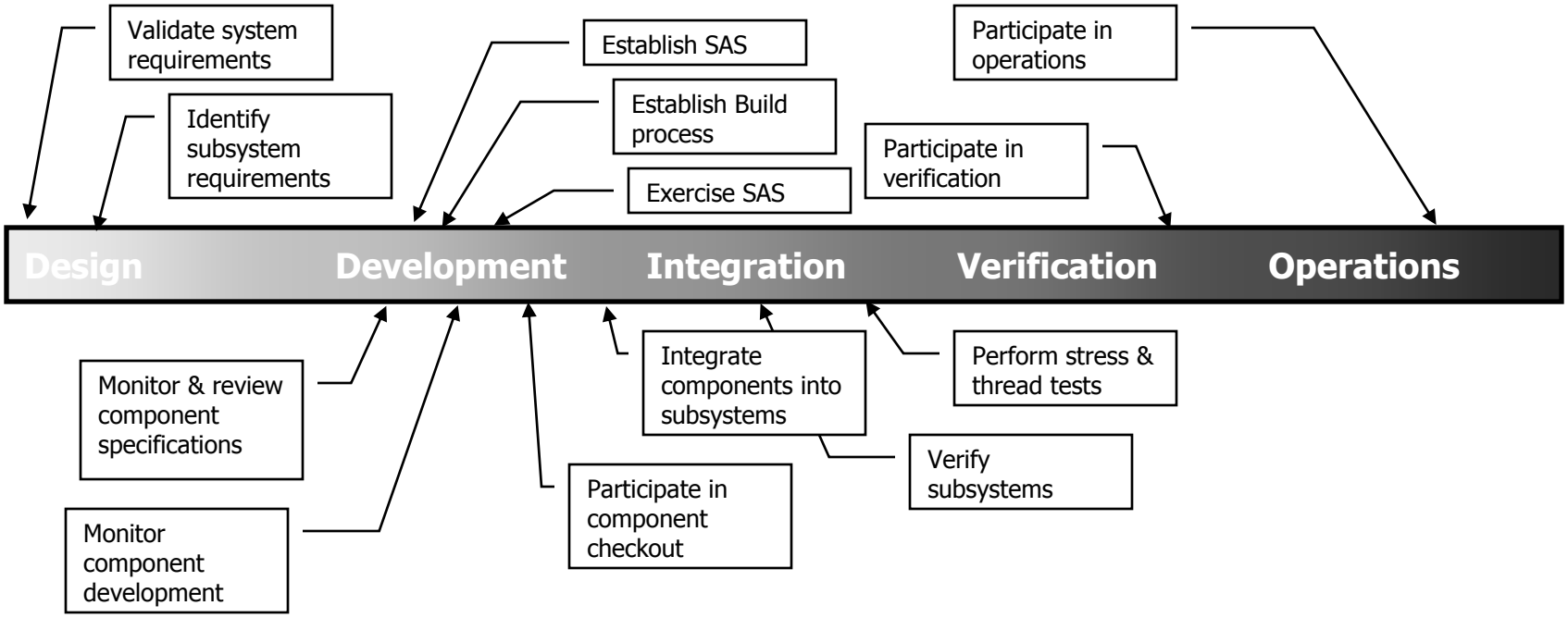
2: Increased Subsystem Emphasis

- A subsystem is an end-to-end functional thread through the system that satisfies a set of system-level requirements for capabilities and performance.
- A subsystem corresponds to a subset of the overall system physical implementation.

Hardware / Software Configuration Items & Components	Subsystems				
	Subsys A	**Subsys B**	**Subsys C**	**SubsysD**	**SubsysE**
HWCI 1	X	X	X	X	X
HWC 1.1	X	X	X	X	X
HWC 1.2	X	X	X	X	X
HWCI 2	X		X	X	X
HWC 2.1	X		X	X	X
HWC 2.2	X		X	X	X
HWC 2.3	X		X	X	X
HWCI 3		X		X	X
HWC 3.1		X		X	
HWC 3.2		X		X	
SWCI 4	X	X	X	X	X
SWC 1.2	X	X	X	X	X
SWC 1.2	X	X	X	X	X
SWC 1.3	X	X	X	X	X
SWCI 5	X				
SWCI 2.1	X				
SWCI 2.2	X				
SWCI 2.3	X				
SWCI 2.4	X				
SWCI 2.5	X				
SWCI 6		X	X		X
SWCI 7			X	X	X
SWCI 4.1			X	X	X
SWCI 4.2			X	X	X

3: System Integration Team Composition

- Sys Integ Team should be composed of Responsible Engineers (REs), each of whom owns one or more subsystems.
- Each RE is responsible for successfully fielding their subsystem's capabilities.
- RE responsibilities encompass entire program, from System Specification through fielding, including all HW and SW components and all suppliers.
- System Integration team uses internal Team funding for integration activities, and uses other program funding when supporting other activities. System Integration also funds engineers who support the System Integration process.



4: System Architecture Skeleton

NORTHROP GRUMMAN

- The SAS provides a benchtop environment that mimics the actual system physical architecture.
  - Commercial hardware can be used to create the SAS when this improves schedule.
  - ESC has criticized our failure to budget for this area.
- The SAS allows early HW & SW infrastructure implementation and checkout.
- The SAS provides an environment for component verification.
- The SAS provides an environment for incremental subsystem integration.
- The SAS also enables early, concrete milestones that can be formally demonstrated to the customer.
  - Allows witnessed subsystem verification before the entire system is in place.
- Systems Integration owns the Systems Integration Lab (SIL) and ensures that the SAS is sufficient to support all users.

5: Configuration Control

NORTHROP GRUMMAN

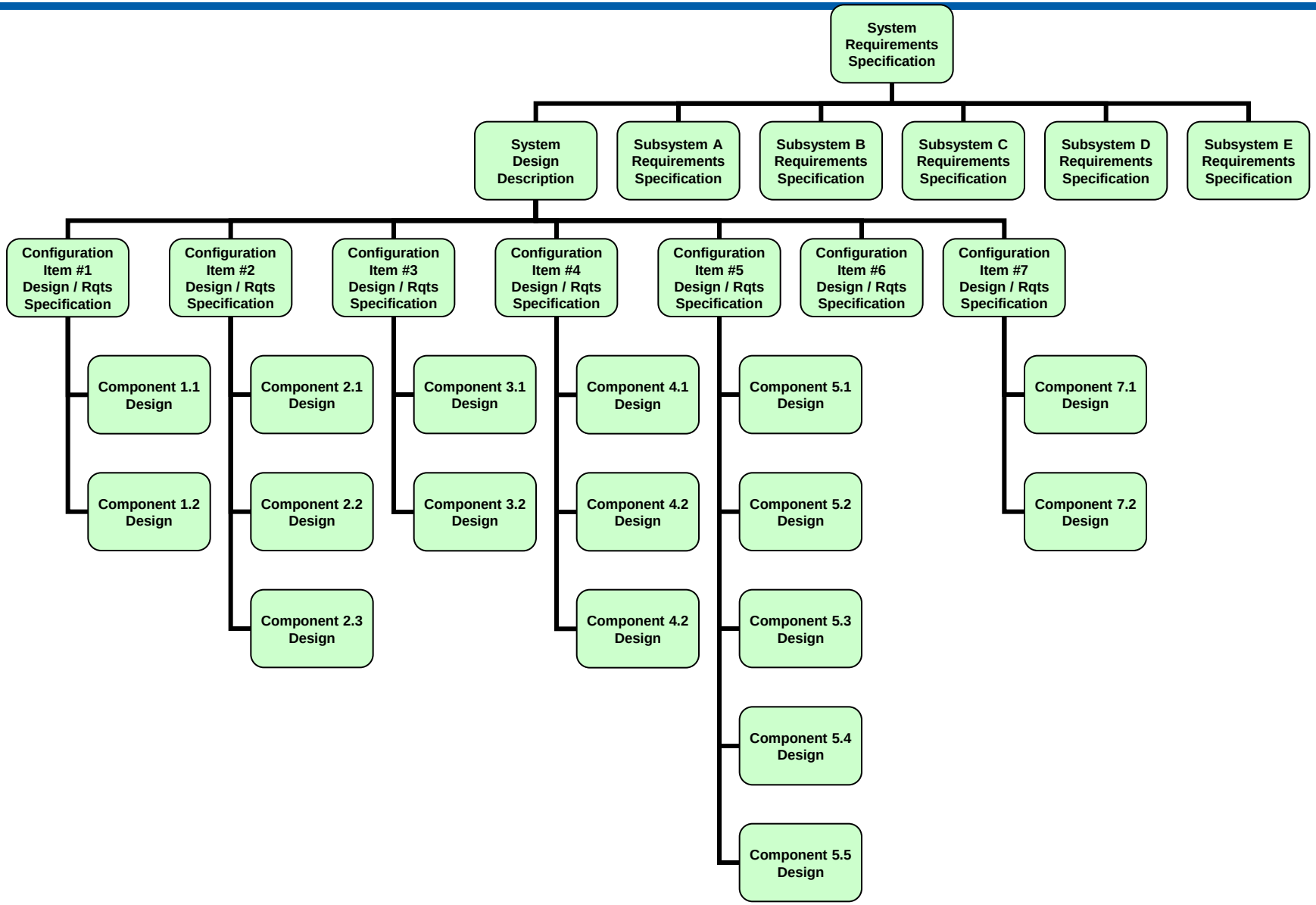
- The Configuration Management and Systems Integration Teams should own the hardware and software configurations.
  - Do not allow the Hardware and Software Teams to own the configuration.
- CM and SI Teams should perform software application builds and system builds using controlled source code.
  - This ensures that all configuration settings & files are controlled.
- CM Team should control hardware drawing flow:
  identification ⇨ design ⇨ check ⇨ review ⇨ release ⇨ fabrication ⇨ correction
- The Environment and Network (EnvNet) Team can be responsible for operating system and network device configuration. Alternatively, this can be allocated to SI Team as an aspect of the SAS initiative. Information Assurance Team has strong influence in either case.
- The SI Team should control the lab configurations and should enforce entry criteria (component integration readiness review) prior to component integration.

6a: Component & Subsystem Specifications

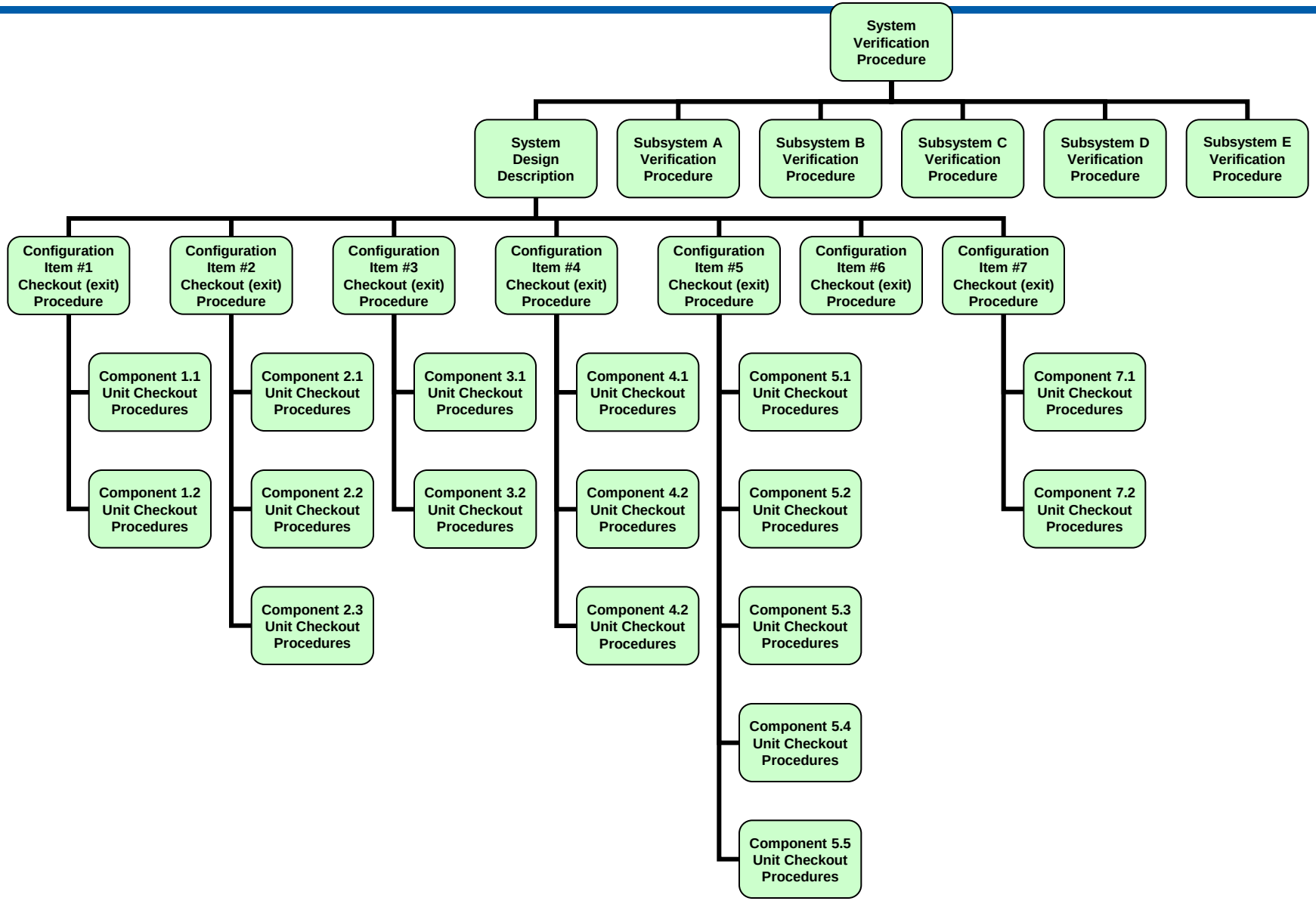
NORTHROP GRUMMAN

- Specifications & procedures are developed in order to reduce integration & rework costs for complex and / or high-risk items.
- Three categories of specifications and procedures:
  - Requirements:
    - System requirements (and their allocation)
    - Derived requirements
  - Design:
    - High level design
    - Detailed design
  - Verification:
    - System requirements verification
    - Component checkout
- Specifications and procedures are like medicine – use the right dose.
  - If an item is simple and low risk then little is needed.
  - If an item is complex and high risk then do more.
  - Goal is to minimize expected program cost through final acceptance, balancing initial budgeted cost against residual risk.

6b: Spec Tree – Requirements & Design



6c: Spec Tree – Verification



7: Component Checkout for Integration Entry

NORTHROP GRUMMAN

- The Software and Hardware / Fabrication Teams are responsible for comprehensive component checkout before releasing components to subsystem integration.
- Software checkout is performed using written procedures and includes:
  - Unit testing – exercise all new software using test drivers when necessary.
  - Component testing – verify component interfaces and compliance with derived requirements; also verifies good practices such as boundary checks, error handling, and memory leak testing.
  - PCR testing – confirm that PCR corrections function as intended.
  - Device integration – verify that device controllers (such as terminal hosts) communicate correctly with the associated devices. This may be done using the SAS environment, but is not considered to be an aspect of subsystem integration.
- Hardware / Fabrication checkout is performed using written procedures and includes:
  - Physical configuration check – audit against the drawings.
  - Cable check – buzz all cables for continuity, shorts, and cross pins.
  - Built in test – exercise any BIT that is inherent to a device.
- When checkout is complete the component is submitted to the SI Team for subsystem integration within the SAS.
  - Software VDD should accompany each new release.
  - Informal Integration Readiness Review may be used to confirm documentation.
- Note that the Continuous Integration process continues independently of the component checkout and release for subsystem integration.

8: Tracking Integration Progress

- Focus is on successful subsystem integration and corresponding system-level requirements verification.
  - Metrics track percent of system requirements that have been verified.
- Also track component release into integration in order to understand dependencies.
  - Metrics track percent of HW and SW released into integration.

Hardware / Software Configuration Items & Components	Subsystems				
	Subsys A	Subsys B	Subsys C	SubsysD	SubsysE
HWCI 1	C	I	C	C	C
HWC 1.1	C	I	C	C	C
HWC 1.2	C	I	C	C	C
HWCI 2	X		X	X	X
HWC 2.1	X		X	X	X
HWC 2.2	X		X	X	X
HWC 2.3	X		X	X	X
HWCI 3		I		C	
HWC 3.1		I		C	
HWC 3.2		I		C	
SWCI 4	C	I	C	C	C
SWC 1.2	C	I	C	C	C
SWC 1.2	C	I	C	C	C
SWC 1.3	C	I	C	C	C
SWCI 5	C				
SWCI 2.1	C				
SWCI 2.2	C				
SWCI 2.3	C				
SWCI 2.4	C				
SWCI 2.5	C				
SWCI 6			C		C
SWCI 7			X		X
SWCI 4.1			X	X	X
SWCI 4.2			X	X	X

Green fill = integrated

Yellow fill = released

9: Continuous Regression Testing

NORTHROP GRUMMAN

- The SAS and all integrated components are continuously tested to detect regression and promote robust performance.
- Focus is on end-to-end system requirements verification, to the extent that is possible within the SAS and the currently integrated subsystems.
- Perform “smoke testing” to provide system component coverage testing and promote robust operations.
  - Use automated test tools to eliminate hands-on testing
  - Test tools may be internal (built in stimulation / check) or external.
  - New integration baselines receive smoke test prior to functional testing.

10: Stress & Mission Thread Testing

NORTHROP GRUMMAN

- Customer satisfaction is unlikely if the system just barely meets the requirements.
- Our customers expects robust system performance in a real operational environment.
- Stress testing includes:
  - Processor and storage measurements in various system conditions.
  - “Corner condition” testing.
  - System operation beyond required system capacities.
  - Endurance testing and “woodpecker” testing.
- Long mission thread testing includes:
  - Operationally relevant scenarios.
  - User testing (focus groups) to ensure suitability.

System Integration Role –Summary

NORTHROP GRUMMAN

- **Entry criteria**
  - Program office authorization based upon contract award.
- **Inputs**
  - System Specification (but Sys Integ staff are cross-assigned to the Sys Eng Team)
  - System Design Description (plus lower level specifications as they are created)
- **Activities**
  - Validate requirements, allocate subsystem requirements, establish continuous integration process, monitor component developments, perform subsystem integration, perform system integration, participate in testing and operations. See previous slides.
- **Reporting**
  - Primarily to program SEIT Lead, who reports to the program office.
- **Signoff**
  - Responsible Engineers are major stakeholders in all peer reviews.
  - Chief Engineer is final approval authority.
- **Responsible parties**
  - Systems Integration Lead, Responsible Engineers, potentially Infrastructure Engineers.
- **Exit criteria**
  - All subsystems integrated; all subsystem & system threads exercised successfully.
  - System is robust and is ready for formal requirements verification.

Backup Materials

System Integration Best Practices

- A Systems Integration Plan is developed during program planning.
- The overall system is defined as Configuration Items matrixed against Subsystems. The WBS, BOM, and Schedule are each organized using these definitions.
- Appropriate Requirements, Design, and Verification specifications are defined at each level of the System.
- A drawing tree identifies Requirements, Design, and Verification documents across the Configuration Items and Subsystems.
- Requirements traceability is maintained between Requirements, Design, and Verification specifications at all levels. Program plans, including development, build, release, and verification schedules, all trace to the specific requirements that are addressed.
- The System Integration planning and management methodology tracks and documents the progress of each Configuration Item (with at least one level of decomposition) and each Subsystem.
- Responsible Engineers are assigned to each Subsystem and are budgeted within Systems Integration throughout the program, with appropriate budget synergy between Systems Engineering, Systems Integration, Development, and Test.
- Configuration Management (CM) owns the system configuration; all items are submitted to Systems Integration through CM.
- A SIL is defined and provides distinct spaces & equipment for Component Checkout, Subsystem Integration, and System Integration.
- A System Architecture Skeleton (SAS) is assembled in the SIL early in the program. It hosts the continuous integration activities and incrementally aligns with the deliverable system as components are delivered to the SIL. The SAS includes simulators when needed in order to exercise system components early in the program.
- A continuous integration rhythm is defined using the System Architecture Skeleton. The integration rhythm includes regular system builds and regression testing. Automated test methods are used in order to cost-effectively conduct frequent “smoke tests”.
- Component checkout using a written procedure is performed for each component prior to Subsystem Integration. Component checkout verifies component interfaces (possibly using simulators) and compliance with derived requirements; it also verifies good practices such as boundary checks, error handling, and memory leak testing
- Subsystem integration using a written procedure is performed for each Subsystem prior to System Integration. Subsystem Integration verified each major internal interface and ensures that end-to-end, system-level capabilities are in place.
- System Integration using a written procedure is performed prior to Verification. System Integration exercises each item (hardware and software) of each configuration item. System Integration includes stress testing and long mission thread testing.

Programs should be audited against this checklist to ensure conformance with these recommendations.

Systems Integration Work Breakdown Structure