Space Shuttle Program
Primary Avionics Software System (PASS) Success Legacy – Major Accomplishments and Lessons Learned
Presentation Summary

James K. Orr

August 24, 2010

United Space Alliance

Agenda

- Introduction
- Dedication To Safety
- Space Shuttle Flight Software Period Themes
- Space Shuttle Flight Software Accomplishments
- Look At Improvements Through Latent Product DRs
- Space Shuttle Flight Quality Measurements
- Space Shuttle Flight Software Reliability
- Space Shuttle Flight Software Lessons Learned
- Summary
- Acronyms

Introduction

- This presentation focuses on the Space Shuttle Primary Avionics Software System (PASS) and the people who developed and maintained this system.
  - One theme is to provide quantitative data on software quality and reliability over a 30 year period
  - Second theme is to focus on the people and organization of PASS
- Quality Measures
  - Pre-build Detection Effectiveness (Inspection Plus Development Test)
    - Errors found by Inspection and Development Test Pre-Build (prior to being placed under project configuration control) divided by total errors
  - Verification Effectiveness
    - Process DRs divided by (Process DRs plus Product DRs)
  - Product Error Rate
    - Product DRs divided by new, changed, deleted source lines of code. Includes only non-comment source lines of code.

Introduction

- Quality Measures
  - Consistent data relates to “code break” discrepancies
    - Requirements were supplied from external sources
  - Errors counted in three periods
    - Errors found by Inspection and Development Test Pre-Build (prior to being placed under project configuration control)
    - Process DRs found Post Build until a milestone called Software Readiness Review (SRR) for the first flight off that increment; typically occurs approximately 4 weeks prior to flight
    - Product DRs found from SRR of first flight until end of program
      - Subset of Product DRs are those which occur in either terminal countdown or in flight, called in-flight DRs
    - Additional special category of DRs are called Released Severity 1 DRs. These may be process or product DRs. These are DRs that could cause loss of crew or vehicle that are released to any field site such as the Shuttle Mission Simulator (SMS), the vehicle at KSC, or the Shuttle Avionics Integration Lab (SAIL).

Introduction

- Common themes running through lifecycle periods
  - Improvements through process enhancements
  - Improvements through automation
  - Defect removal following identification of significant process escapes
  - Impact of workforce instability
  - Early evaluator, adopter, and adapter of state-of-the-art software engineering innovations
- A significant contributor to the success of the PASS FSW organization has been the support of the NASA PASS software customers that have consistently valued quality and supported reasonable implementation schedules. NASA has also supported maintaining critical skill staffing.

Dedication To Safety

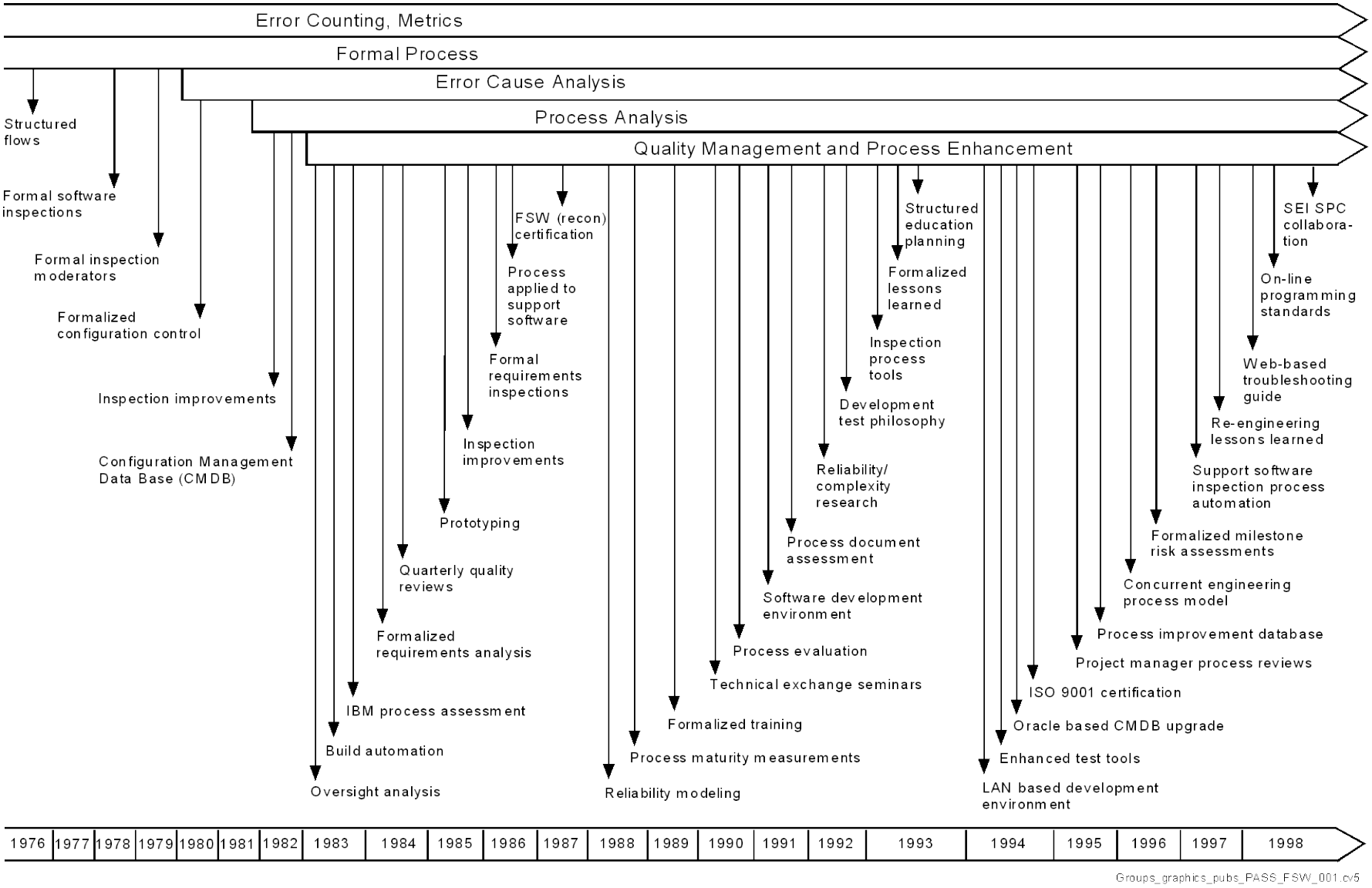
Dedication To Safety

- Developing complex human-rated flight software is a major technical challenge.
  - Perfection required to achieve the desired level of safety
    - Extremely difficult to accomplish, but can be aggressively pursued
  - Keys to the pursuit of perfection
    - Principles of Providing High Reliability Software
    - Continuous Process Improvement
    - Defect Elimination Process

Principles of Providing High Reliability Software

- Safety certification is currently based on process adherence rather than product.
- Assumption is that a known, controlled, repeatable process will result in a product of known quality.
- Process executed by personnel that are *committed to safety and skilled* relative to processes, system architecture, and specialized software requirements.
- Team skills and workload closely monitored by management to prevent over commitment that could result in quality breakdowns.
- Use “trusted” tools to develop, build, release and maintain the software.
- Use measurements to continuously assess the health of both the process and the product.
- Relationship between quality and reliability must be established for each software version and statistically demonstrated for the required operational profiles.
- Quality must be *built into* the software, at a *known* level, rather than *adding* the quality after development.
  - You cannot *test* quality into software

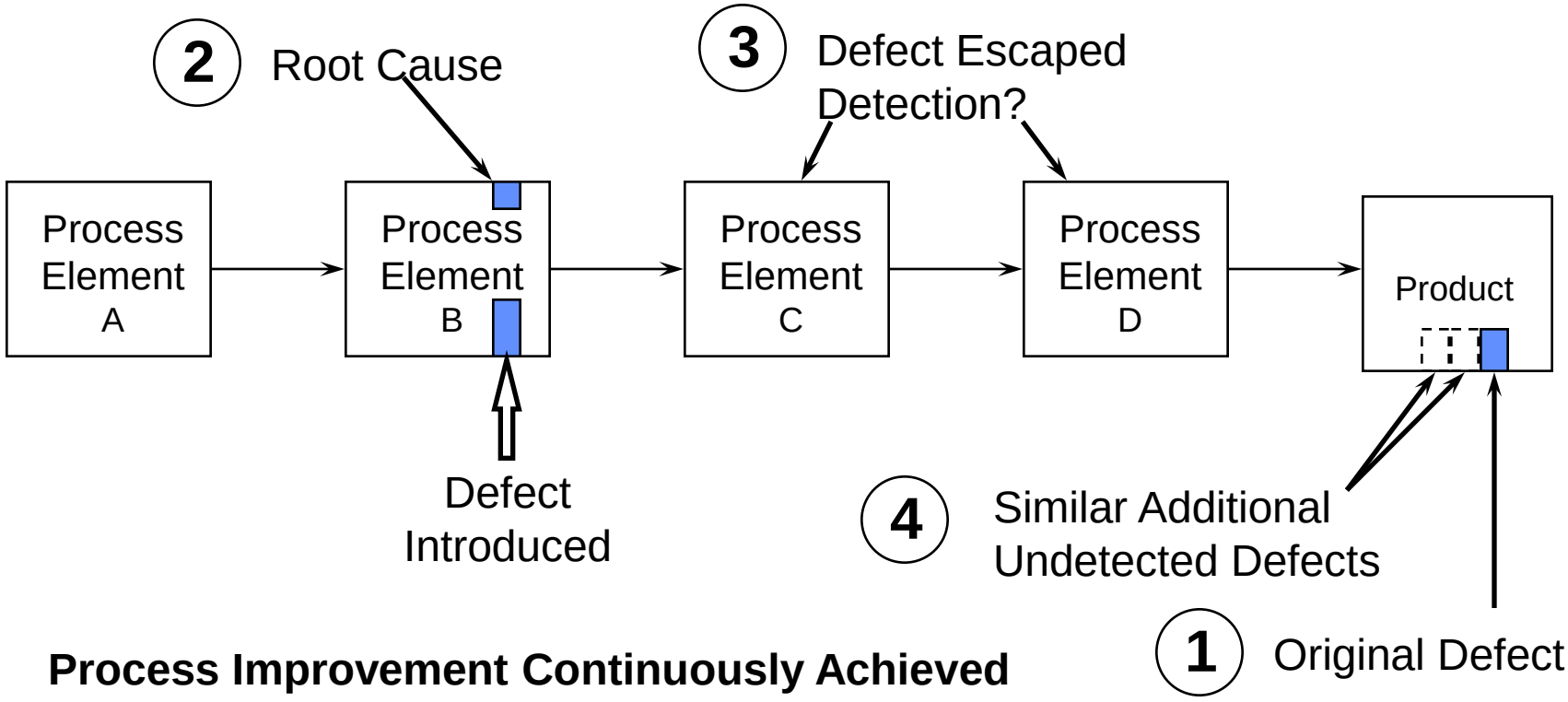
Examples Of Continuous Process Improvement



Defect Elimination Process

Steps performed

1. **Remove defect**
2. **Remove root cause of defect**
3. **Eliminate process escape deficiency**
4. **Search/analyze product for other, similar escapes**



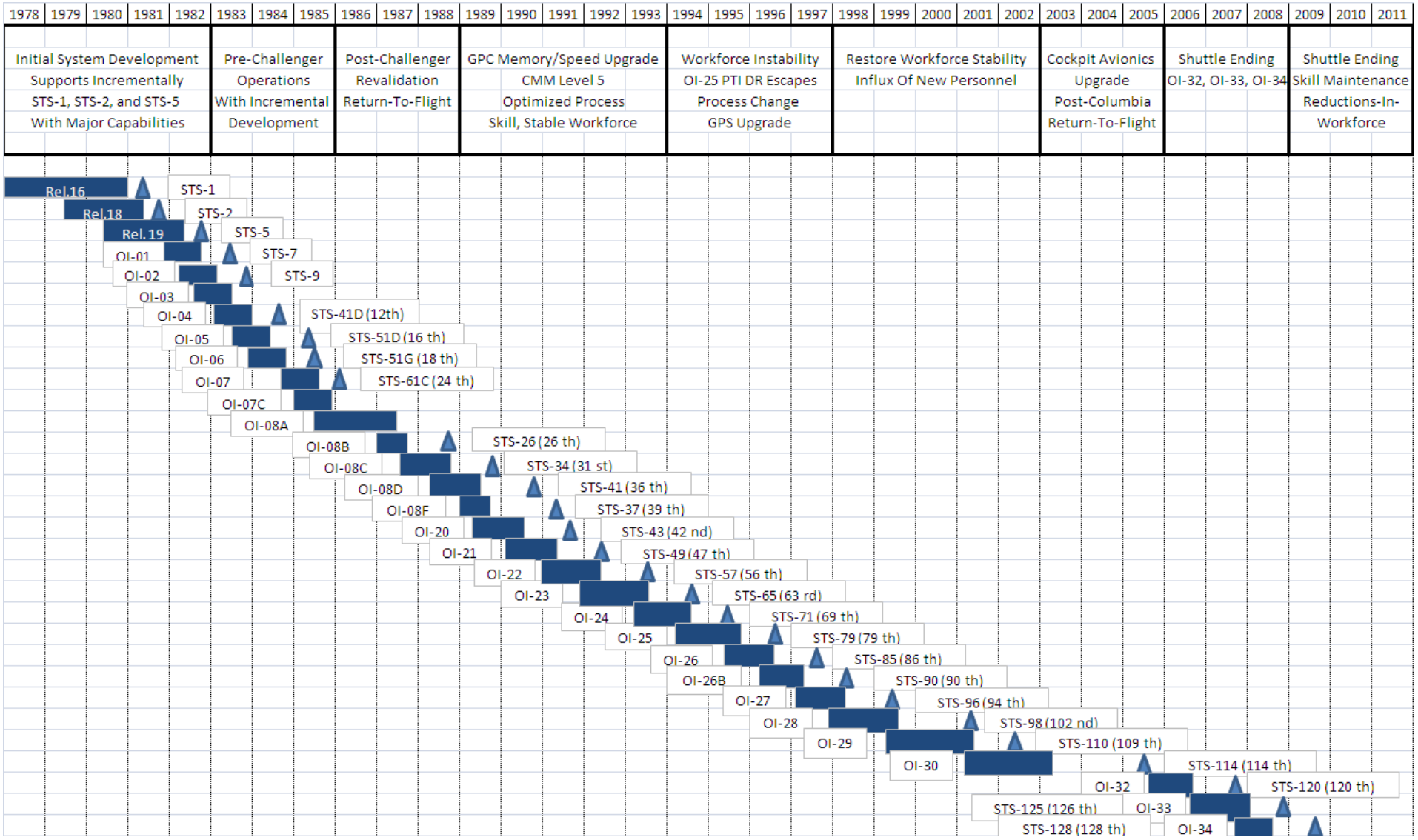
Process Improvement Continuously Achieved By Performing Feedback Steps 2 and 3

Space Shuttle Flight Software Period Themes

Space Shuttle Flight Software Period Themes

Years	Theme	Events
1978-1982	Initial System Development	Supports Incrementally / STS-1 to STS-5 Many Major Capabilities
1983-1985	Pre-Challenger Operations	Incremental Development / Reductions in Staff during 1985
1986-1988	Post-Challenger, Return to Flight	Challenger Accident / PASS FSW Revalidation / Return to Flight
1989-1993	Process Optimization and Stability	CMM Level 5 / GPC Memory/Speed Upgrade / Skilled, Stable Workforce
1994-1997	Transition To Loral / Lockheed Martin	Workforce Instability / OI-25 PTI DR Escapes / GPS Upgrade
1998-2002	Transition to United Space Alliance	Restore Workforce Stability / Influx Of New Personnel
2003-2005	Post-Columbia / Return-To-Flight	Cockpit Avionics Upgrade / Columbia Accident / Return to Flight
2006-2008	Shuttle Ending, OI Development	OI-32, OI-33, OI-34 / Display Upgrades evolved From CAU / CMMI Level 5 November 2006
2009-2011	Shuttle Ending, Skills Maintenance	Skills Maintenance / Reductions-In-Workforce / CMMI Level 5 in September 2009

PASS FSW Development History



Space Shuttle Flight Software Accomplishments

Initial Development (1978 – 1982)

Category	Observation / Characteristics
Scope Of Development	▪ Transition from ALT work to OFT development ▪ Expansion of orbit FSW capability post STS-1 ▪ First Flight Capabilities ▪ Schedule driven, heavy change request traffic ▪ Early Systems Management / Payload Management Software

Pre-Challenger Accident (1983 – 1985)

Category	Observation / Characteristics
Scope Of Development	▪ Rendezvous ▪ Full Redesigned SM/PL Capabilities ▪ RMS Deploy and Retrieval ▪ Centaur Development ▪ Spacelab ▪ Main Engine Control redesign ▪ Payload manifesting flexibility ▪ Crew enhancements ▪ Enhanced ground checkout ▪ Western Test Range (Vandenberg) ▪ Reconfiguration tool planning / development ▪ Tools and procedures planning / development for DOD flights

Post Challenger Accident (1986 – 1988)

Category	Observation / Characteristics
Scope Of Development	▪ Post-51L Safety Changes ▪ Bailout Capability ▪ Abort Enhancements

CMM Lvl 5 Process Under IBM (1989 to 1993)

Category	Observation / Characteristics
Scope Of Development	▪ GPC Upgrade ▪ Extended Landing Site Table ▪ OPS 3 (TAL Code) in upper memory ▪ Redesigned Abort sequencer ▪ 2 Engine Out Auto Contingency Aborts ▪ OV-105 Hardware changes ▪ On-Orbit Changes ▪ MIR Docking ▪ On-Orbit DAP Changes

Loral / Lockheed Martin (1994 to 1997)

Category	Observation / Characteristics
Scope Of Development	▪ Mir Docking Adapter ▪ On-Orbit DAP Changes ▪ 3 Engine Out Auto Contingency Aborts ▪ Ascent Performance Enhancements ▪ Single-String GPS

United Space Alliance (1998 to 2002)

Category	Observation / Characteristics
Scope Of Development	▪ 3-String GPS ▪ East Coast Abort Landing (ECAL) Automation ▪ Automatic Reboost ▪ GPC Payload Command Filter (GPCF) ▪ Increased data to MEDS ▪ Start of Cockpit Avionics Upgrade (CAU) builds

Post-Columbia, Return To Flight (2003 to 2005)

Category	Observation / Characteristics
Scope Of Development	▪ Last of CAU builds ▪ Enhanced ADI / HSI capability

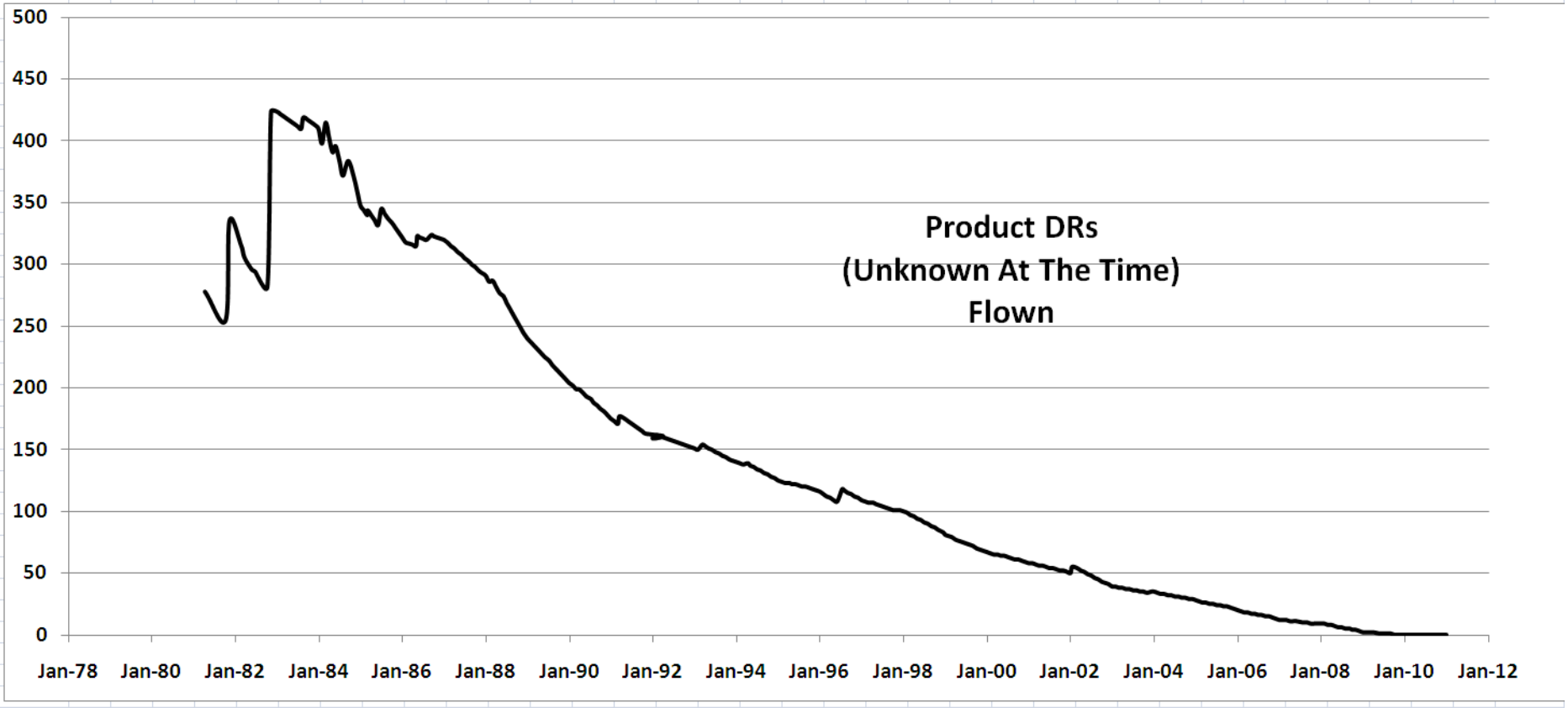
OI Development Continuing (2006 – 2008)

Category	Observation / Characteristics
Scope Of Development	▪ Lambert Guidance Improvements ▪ 6x Traj display redesign ▪ Entry and Ascent Bearing Display additions ▪ RTLS ET Sep improvements ▪ Entry Remote Controlled Orbiter (RCO) Capability ▪ Elimination of old user notes and DRs ▪ Reduction in Horizontal Sit display code size ▪ Year End Roll Over (YERO)

Look At Improvement Through Latent Product DRs

Number Of Latent Unknown Product DRs Flown

1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011
Initial System Development Supports Incrementally STS-1, STS-2, and STS-5 With Major Capabilities					Pre-Challenger Operations With Incremental Development			Post-Challenger Revalidation Return-To-Flight			GPC Memory/Speed Upgrade CMM Level 5 Optimized Process Skill, Stable Workforce					Workforce Instability OI-25 PTI DR Escapes Process Change GPS Upgrade					Restore Workforce Stability Influx Of New Personnel					Cockpit Avionics Upgrade Post-Columbia Return-To-Flight			Shuttle Ending OI-32, OI-33, OI-34			Shuttle Ending Skill Maintenance Reductions-In-Workforce		
1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011



Product DRs that existed on a flown system, but were unknown at the time of the flight . Discovered up to 25 years later.

Summary Of Product DR By Period

Years	New Product DRs Introduced In Period	Latent, Unknown Product DRs At End of Period	Flight Days Over Period	DRs Occurring In Flight	Latent, Unknown Severity 1 DRs
1978-1982	523	424	29	3	4
1983-1985	109	322	147	8	6
1986-1988	16	240	None	No Flights	No Flights
1989-1993	22	140	291	1	0
1994-1997	12	100	365	4	0
1998-2002	8	39	675	2	0
2003-2005	0	20	None	No Flights	No Flights
2006-2008	1	2	162	1	0
2009-2011	0	N/A	114	0	0

- Supporting information available in more detail backup presentation.

Take Away From History Of Latent Product DRs

- Initial System Development period for future systems likely to be similar to that for the Space Shuttle
  - Likely first flight will have a large number of unknown software discrepancies some of which may be Severity 1
- Transition to Operations period likely to produce the maximum number of in-flight software discrepancies
  - Significant software maintenance activity as latent Product DRs are discovered, analyzed, and fixed as appropriate
  - Concurrently, program push to add capabilities which could not be implemented during the development period
  - Staffing reduction occurring from development period level to steady state operations level
  - Planned automation likely slightly behind staffing reductions based on the original automation schedule
- Quality improvements made in 1986 to 1988 for PASS were critical to the success in later years
- It takes a long time for unknown latent software defects to be identified

Space Shuttle Flight Software Quality Measurements

Summary Of Quality Metrics By Period

Years	Product Error Rate DRs / KSLOC	Pre-Build Error Detection Effectiveness	Verification Effectiveness (Percent Found By SRR)	Notes
1978-1982	0.8 (STS-1) to 1.1	Information Not Available	77 % to 91 % (STS-1)	
1983-1985	2.8 (OI-1) to 1.1	40 % to 65 %	70 % to 80 %	Very Short Cycle - Release Every 4 Mo.
1986-1988	0.7 to 0.2 (OI-8C)	Near 80 %	60 % to 70 %	Return-to-flight Critical Changes
1989-1993	0.1 to 0.2	80 % to 90 %	80% to 90 %	
1994-1997	0.1 to 0.2 except 0.8 for OI-25	75 % to 85 %	85 % to 100 % except 60 % for OI-25	Isolated Process Escape on OI-25
1998-2002	0.1 to 0.2	85 % to 90 %	85 % to 95 %	
2003-2005	CAU Canceled	CAU Canceled	CAU Canceled	Work on CAU required changes, Later CAU Canceled
2006-2008	0.0 to 0.1	80 % to 100%	95 % to 100 %	
2009-2011	No OI Development	No OI Development	No Development	Reduced Flight System Changes Only, No OI Dev.

Take Away From History Of Quality Metrics

- STS-1 quality was extremely high
  - 0.80 Product DRs / KSLOC
  - 91 % of DRs detected by flight
- Transition to operations in parallel with shortened development schedules (CI every 4 months) resulted in a sharp increase in Product Error Rate
  - OI-1, Product Error Rate 2.8 Product DRs per KSLOC
- By OI-8C (1988) Product Error Rate down to 0.2 Product DRs per KSLOC
  - Range of 0 to 0.2 Product DRs per KSLOC has been sustained since then with the exception of one capability on OI-25
- Significant effort expended on PASS changes required for Cockpit Avionics Upgrade (CAU). PASS changes not used following cancellation of CAU.

Space Shuttle Flight Software Reliability

Summary Of Modeled Reliability By Period

Years	Calendar Days Between Any Product DR	Flight Days Between In-Flight DRs	Risk To Shuttle Due To Severity 1 FSW DR
1978-1982	6 (STS-1), 7 (STS-5)	7 (STS-1), 9 (STS-5)	1 in 327 (STS-1) to 1 in 409 (STS-5)
1983-1985	10 to 19	12 to 24	1 in 552 to 1 in 1072
1986-1988	29 at STS-26	90 at STS-26	1 in 1599 at STS-26
1989-1993	29 to 42	90 to 131	1 in 1599 to 1 in 2335
1994-1997	42 to 54	131 to 120	1 in 2335 to 1 in 3161
1998-2002	54 to 61	120 to 140	1 in 3161 to 1 in 3491
2003-2005	75 at STS-114	235 at STS-114	1 in 4212 at STS-114
2006-2008	75 to 88	235 to 276	1 in 4212 to 1 in 4930
2009-2011	88 to 94	276 to 294	1 in 4930 to 1 in 6260

- Risk level of approximately 1 in 1000 at January 2006 established during reliability research in the late 1980's as a Return-To-Flight action. Variation over time based on MTBF as calendar days between any Product DR's.

Take Away From Reliability Information

- Early failure rates (calendar days between Product DRs) were on the order of 10 days.
  - Almost identical in-flight failure rate (flight days between in-flight DRs)
- Trend of risk of Severity 1 DR (loss of crew or vehicle)
  - Early flights on the order of 1 in 400
  - STS-26 on the order of 1 in 1600
  - Current flights on the order of 1 in 6000
    - Severity 1 DR small sample size insufficient to accurately assess probabilities
    - Estimates are believed to be conservative (actual risk is lower)
- Extraordinary improvement in-flight failure rate from STS-51L (Challenger Accident) to STS-26 (Return-to-Flight)
  - From 24 flight days between in-flight DRs to 90 days (factor of 3 increase)
  - Major contributors
    - “Revalidation” Return-to-flight audits and actions
    - Full implementation of automatic flight reconfiguration
    - Factor of 3 increase in Software Production Facilities testing capacity driven by AP-101B/AP-101S concurrent production

Space Shuttle Flight Software Lessons Learned

- This section provides summaries of key items for lessons learned from in-flight DRs and released Severity 1 DRs.
- Data from across all periods was developed separately. Then, lessons learned were consolidated.

Lessons Learned Over the Years

- Scenarios
  - All possible scenarios must be identified, addressed in requirements, accommodated via design/code, and tested.
    - Insure that proper initialization will occur under all scenarios supported by the software.
    - Requirements must address what is to be done for failed hardware under all scenarios supported by the software.
    - Scenario analysis must identify the maximum ranges for parameters under all scenarios. Variable precision must correctly support the maximum ranges.
    - Many scenarios-related problems have extremely small timing windows. Very unlikely to detect during testing only. May require “Multi-Pass” analysis methods to insure identification.
    - Implement robust scenario testing. Adequate test facility resources required including resources for off-nominal testing.

Lessons Learned Over the Years

- Interface Testing
  - Software Interface Control Document requirements must be explicitly verified in an end-to-end manner.
    - Two in-flight DRs due to failure to verify PASS SM to Spacelab ICD.
      - Both required in-flight patches due to mission objective impacts for specific payloads.
- Analyze Error Logs
  - Anomalies occurring which resulted in signatures being written to software error logs may go undetected unless the software error logs are analyzed after the run.
    - Procedures are documented to always require analysis of software error logs after completion of the test.
    - Test may be designed to automatically stop if a certain types of error conditions occur such as GPC errors.
      - Allows for capturing full software memory at the time of the error condition.

Lessons Learned Over the Years

- Simulation Models
  - Collect appropriate data during simulated hardware tests so that any anomaly occurring is identified.
  - Models in the software test environment must provide valid outputs
    - Timing related software problems will be impossible to detect in simulation unless the hardware models provide random variation similar to actual hardware characteristics.
    - Using checkpoints and restart capabilities may additionally limit the number of opportunities to observe timing related software problems even if the simulation supports random hardware timing variation.
      - Use of checkpoints and restart capabilities is highly desirable. These capabilities allow for efficient use of test facility resources. These capabilities also greatly enhance the ability to duplicate problems by providing a duplicate software/environment state just prior to the event being duplicated.

Lessons Learned Over the Years

- Hardware / Software Integration Testing
  - Using checkpoints and restart capabilities may limit the number of opportunities to observe timing related software problems.
    - Multiple tests from the same source checkpoint will have one fixed set of software internal timing relationships.
  - Collect appropriate data during hardware tests so that any anomaly occurring is identified.
  - Latent defects can remain in the FSW multiple years until scenario and hardware re-action timing align .
- Manual processes
  - Many process such as late updates to flight reconfiguration are initially done manually. Three in-flight DRs in the 1983 -1985 period introduced this way. Manual processes require continuous management oversight to insure rigorous and correct execution.

Lessons Learned Over the Years

- "Apparently Unrelated" Changes
  - Multiple "apparently unrelated" changes can collectively produce unexpected erroneous consequences.
    - Regression testing required to insure software functions continue to work correctly.
  - Ever present risk to "stumble" into maintenance traps once the maintenance trap is introduced into the software.
    - Maintenance trap is typically the result of waiving programming standards for some small short term savings in implementation time, code space, or computer performance.
    - These short term savings are insignificant compared to the long term project cost incurred during maintenance phase.

Lessons Learned Over the Years

- Essential to formalize management and lead analysts responsibility for assessing skills proficiency and work performance history for every individual on every team and evaluate risk based on skills mix with closed loop responsibility to program manager.
  - Every change to human rated flight software must be implemented with the same professional attention to detail by knowledgeable and motivated personnel.
- Essential to collect measurements data and proactively analyze data to search for “in process” symptoms such as Pre-Build Errors found in inspections by only one inspector.

Lessons Learned Over the Years

- Specific Examples
  - Verification analyst participation in the pre-build inspection process significantly adds quality
    - Prior to mid part of Release 19 (STS-5), the Verification analysts did not participate in pre-build design/code inspections. However, they did participate in inspections of patches implemented on STS-1 due to the perceived increased risk of patch implementation over source changes.
    - Assessment of the quality of the STS-1 patches versus the STS-2 source changes for the same DR and CR implementation resulted in the observation that the STS-1 patches were of higher quality.
    - Following this conclusion, the pre-build design/code inspection process was modified to require participation of the Verification analyst.

Lessons Learned Over the Years

- Specific Examples
  - Sequential inspections (e.g., development only peer review followed by pre-build inspection) are equally effective in removing the same % of errors that exist at the start of the inspection.
    - A single inspection removes about 55 % of errors
    - Two sequential inspection each remove about 55 % of errors remaining at the start of each inspection.
    - Collectively, they remove 80 % of the errors present at the first inspection.
  - Similar results are obtained when the criteria for having a re-inspection generally results in re-inspections when undetected errors remain after an initial inspection
    - With appropriate re-inspection criteria, it is possible to remove 80 % of the errors present at the first inspection.

Summary

Contributors To PASS FSW High Quality

Contributor To PASS FSW High Quality	Context
Multiple releases and multiple iterations of testing prior to STS-1.	Delays in launch date due to TPS and SSME issues provided more testing time and more opportunities to fix identified problems.
Fully automated Flight-to-Flight Reconfiguration Process and Tools	Early flights had a number of System Management in-flight failures due to late manual updates.
Structured “PASS Revalidation” activities between Challenger accident and STS-26	Direct contributor to eliminating Severity 1 (Loss of crew/vehicle) DRs from PASS
Continual enhancements of the Requirements/Design/Code/Test Inspection Processes	▪ Have appropriate participation in each type of inspection including external community participation ▪ Having appropriate re-inspection criteria
Adequate test facility functionality and capacity (equipment to execute cases on flight equivalent hardware)	Significant improvement in in-flight reliability between STS-51L and STS-26 during a period when test facility capacity increased by a factor of 3.
Defined criteria for selection of personnel for teams; define how to resist over commitment of critical skills.	Critical skills management has always been a priority. Re-enforced by action From OI-25 PTI DRs where team skill and over commitment were contributing factors.
Rigorous configuration management of all products including requirements, design, code, and tests.	Basic necessary condition

Wrap-up

- This presentation has shown the accomplishments of the PASS project over three decades and highlighted the lessons learned.
- Over the entire time, our goal has been to
  - Continuously improve our process
  - Implement automation for both quality and increased productivity
  - Identify and remove all defects due to prior execution of a flawed process in addition to improving our processes following identification of significant process escapes
- Morale and workforce instability have been issues, most significantly during 1993 to 1998 (period of consolidation in aerospace industry).
- The PASS project has also consulted with others, including the Software Engineering Institute, so as to be an early evaluator, adopter, and adapter of state-of-the-art software engineering innovations.

Acronyms

Acronyms

	Acronym
ADI	Attitude Direction Indicator
ALT	Approach and Landing Test
AP-101B	Initial flight computer for Space Shuttle; 104 K 32-bit full works of Memory
AP-101S	Upgrade flight computer for Space Shuttle; 256 K 32-bit full words of Memory (256K 32-bit FWs = 1MB 8-bit bytes).
ATV	Automated Transfer Vehicle
CAIL	CEV Avionics Integration Lab
CAU	Cockpit Avionics Upgrade
CEV	Crew Exploration Vehicle
CI	Configuration Inspection
CM	Configuration Management
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integrated

Acronyms

	Acronym
CPU	Central Processing Unit
DAP	Digital Auto Pilot
DOD	Department of Defense
DR, DRs	Discrepancy Report(s)
ECAL	East Coast Abort Landing
ET	External Tank
FSW	Flight Software
GPC	General Purpose Computer
GPCF	GPC Payload Command Filter
GPS	Global Positioning System
HIS	Horizontal Situation Indicator
HTV	H-II Transfer Vehicle
ICD	Interface Control Document
KSC	Kennedy Space Center

Acronyms

	Acronym
KLSOC	1000 Non-Comment Source Lines of Code (new, changed, and deleted)
MEDS	Multifunction Electronic Display System
MIR	Name of the Russian Space Station
MTBF	Mean Time Between Failures
NASA	National Aeronautics and Space Administration
OFT	Orbital Flight Test
OI	Operational Increment
OPS	Operational Sequences
OV	Orbiter Vehicle
PTI	Program Test Input
RCO	Remotely Controlled Orbiter
RMS	Remote Manipulator System
RTLS	Return-To-Launch-Site

Acronyms

	Acronym
RTLS	Return-To-Launch-Site
SAIL	Shuttle Avionics Integration Laboratory
SASCB	Shuttle Avionics Software Control Board
SEI	Software Engineering Institute
SM	Systems Management
SM/PL	Systems Management/Payload
SMS	Shuttle Mission Simulator
SRR	Software Readiness Review, typically 4 weeks prior to flight
SSME	Space Shuttle Main Engine
STS	Space Transportation System
TAL	Transoceanic Abort Landing
TPS	Thermal Protection System
Traj	Trajectory
YERO	Year End Roll Over