



NDIA

**CMMI Technology Conference and User
Group**

NASA Experience with CMM and CMMI

November, 2010

John C. Kelly & Tim Crumbley

Office of Chief Engineer



NASA Experience with CMM and CMMI

- **Outline**
 - **NASA's experience with CMMI model**
 - **NASA's CMMI requirement**
 - **NASA's lessons learned and key impacts from using CMMI**

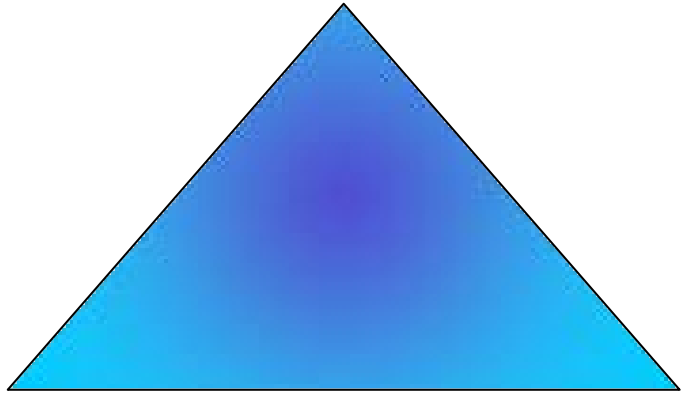


Atop twin columns of fire, space shuttle Atlantis roars into the cloudy sky above Launch Pad 39A at NASA's Kennedy Space Center in Florida on the STS-125 mission.

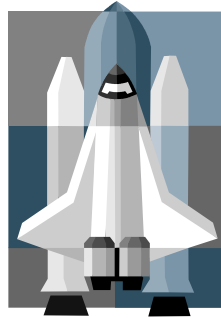


The Three Elements of Project Success

Process:
a defined method involving steps or operations



People:
Skills, Training,
Management



Technology:
Application domains,
tools, languages,
information, environments

**Improved Process + Competent Workforce + Appropriate Technology
=
Reduced Risk, Higher Productivity, and Better Quality**



Timeline 2000 – 2011

NASA Software Engineering Initiative – CMM and CMMI Activities

CMMI Level 3

MSFC
JPL

LaRC*
(FSSB)

LaRC*
(SDAB)

MSFC
(FSW)
JPL

Planned for
FY11
LaRC*
(FSSB)
LaRC*
(SDAB)

CMMI Level 2

MSFC

GSFC
LaRC* SDAB
LaRC* ASDC

GSFC
LaRC
(FSSB)

JSC
KSC
LaRC

MSFC (SIL)
ARC
GRC

GSFC

CMM Level 3

MSFC
ARC*

JSC

CMM Level 2

MSFC

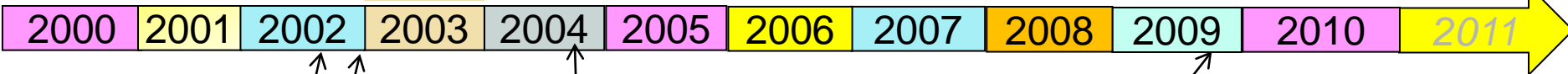
JSC

LaRC
JPL
ARC

GRC
JPL

JPL
JSC

*- Implemented
part of the model



Software Working Group Charter

Started: NASA SW Eng Improvement Initiative

Signed: Software Procedural Requirement NPR 7150.2

NPR 7150.2 (2004)

Class A and B - CMMI Level 2 or CMM Level 3
Class C - Center Decision

5 Centers had experience using the models

Signed: Software Procedural Requirement NPR 7150.2A

NPR 7150.2 A (2009)

Class A - CMMI Level 3
Class B - CMMI Level 2
Class C - Center Decision

8 Centers have experience using the CMMI model

CMM/CMMI Appraisals & Consultations at NASA Centers

Software Engineering Training and SEI Training at NASA Centers



NASA CMMI Summary

Completed SW Engineering Appraisals from FY07-FY10

CMMI = Capability Maturity Model Integrated (Carnegie Mellon University – SW Engineering Institute)

Center/Organization	Rating (SCAMPI A by Certified Appraiser)	Date	# Projects	Type	Organizational size	Software Classes Assessed
LaRC- ASDC	PP(CL3), CM(CL1)	Nov-06	1	Data Center Support	85	Class C
MSFC	ML3	Apr-07	3	Development	63	Class A, B and C
JPL	ML3	Sep-07	7	Dev & Maintenance	1000	Class A, B and C
GSFC	ML2 + RSKM(2)	May-08	4	Dev & Maintenance	600	Class A, B and C
LaRC- FSSB	ML2 + CL3	Oct-08	3	Services	5	Class B & C
LaRC- SDAB	PP(CL3), REQM(CL3), CM(CL3), MA(CL3)	Mar-09	4	Development	21	Class B & C
JSC	ML2	Apr-09	4	Development	90	Class A, B, C and D
KSC	ML2	Sep-09	1	Development	225	Class A, B and C
MSFC – SIL	ML2 + CL3	May-2010	1	Development	50	Class C
ARC –ISD (Code TI)	ML2	May-2010	6	Development	63	Class B & C
GRC-FIt SW	ML2	Aug 2010	2	Development	22	Class C & D
MSFC-FIt SW	ML3	Aug 2010	1	Development	75	Class A
JPL – Mission SW	ML3	Sept 2010	9	Development & Maintenance	950	Class B and C



Sample of NASA Industry Partner Ratings

NASA Projects	Industry Partners	CMMI Level
Shuttle Primary Avionics Software System (PASS), Shuttle SAIL test facility, Orion Crew Exploration Vehicle (Orion)	United Space Alliance Flight Software Element (FSWE)	Level 5
International Space Station (C&DH), Ares	Boeing	Level 3
Orion Crew Exploration Vehicle (Orion)	Lockheed Martin Corporation	Level 3
Orion Crew Exploration Vehicle (Orion), Ares	Honeywell	Level 3
Orion Crew Exploration Vehicle (Orion)	ATK	Level 3
Ares	Draper	Level 3
Ares J-2X, Orion Crew Exploration Vehicle (Orion), Space Shuttle Main Engine	Hamilton Sundstrand Rocketdyne / Pratt & Whitney Rocketdyne	Level 3
Ares, Deep Impact 1	Ball	Level 3
James Webb Space Telescope	Northrop Grumman	Level 3
GRAIL, Juno	Lockheed-Martin Space and Exploration Systems	Level 3
Ground Systems Engineering (GSE) Checkout, Assembly and Payload Processing Services (CAPPS) Kennedy Space Center (KSC)	Boeing	Level 3
ISS Environmental Control and Life Support Systems, Orion Crew Exploration Vehicle (Orion)	Hamilton Sundstrand	Level 3
GOES-R	Harris IT Services Corporation	Level 3
MSFC Engineering Support Contractor	Jacobs Engineering	Level 3
STEREO	Johns Hopkins University Applied Physics Laboratory	Level 3
NASA Software IV&V Services, JSC Support Contractor	L-3 STRATIS	Level 3
KSC Support\Shuttle support	United Space Alliance, LLC	Level 3
NASA Aircraft Management Information System (NAMIS) software	SAIC, Aircraft Operations Support System (AOSS)	Level 3
JSC Support Contractor	Tietronix	Level 2

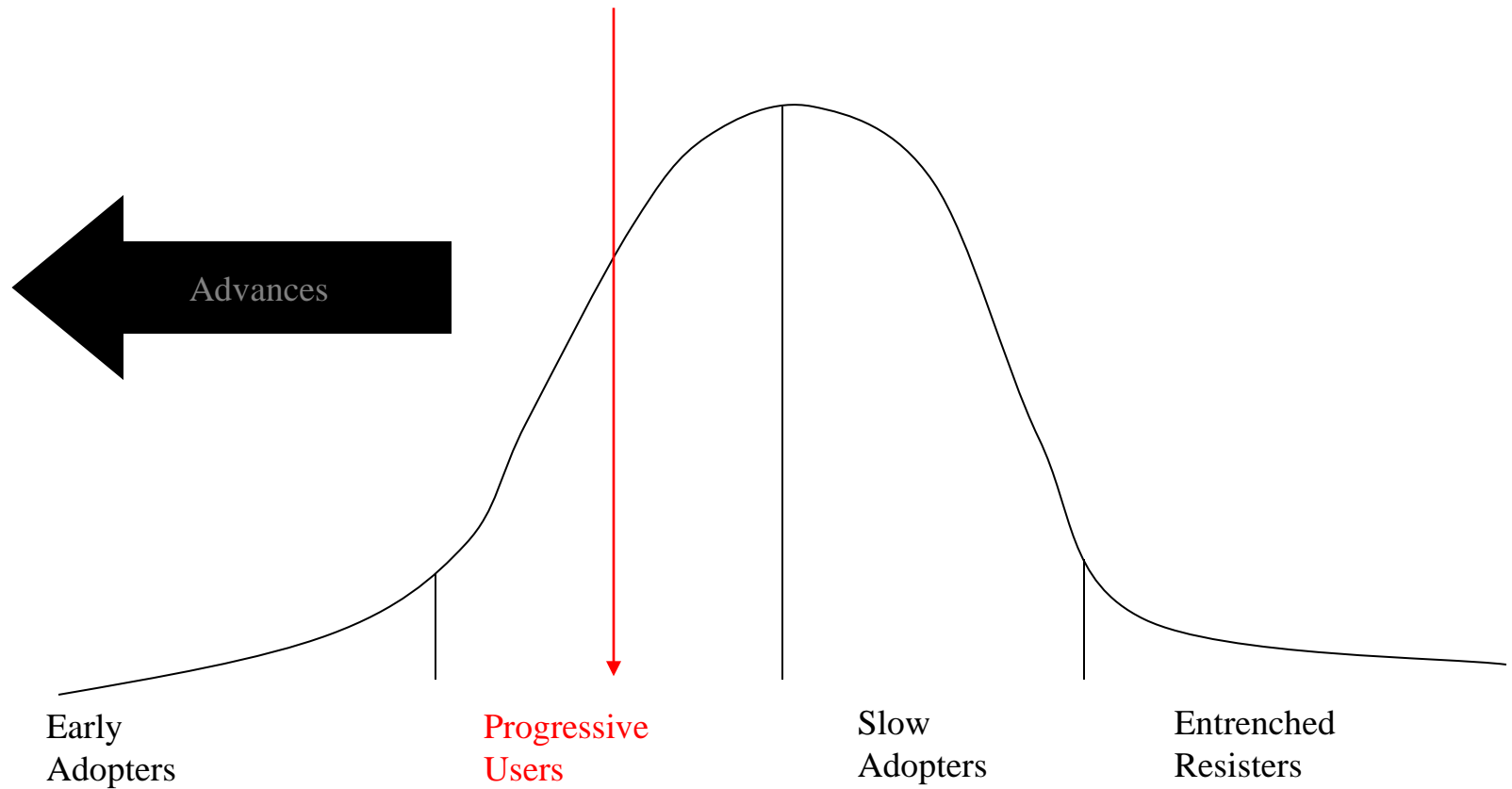


NASA's Software Engineering Requirements

- Software engineering is a core capability and a key enabling technology for NASA's missions and supporting infrastructure.
- The NASA Software Engineering Procedural Requirements (NPR 7150.2A) supports the implementation of the NASA Policy Directive (NPD) 7120.4, NASA Engineering and Program/Project Management Policy.
- The NASA Software Engineering Requirements provide a minimal set of requirements established by the Agency for software acquisition, development, maintenance, retirement, operations, and management.
- The NASA Software Engineering Requirements are intended to support NASA programs and projects to accomplish their planned goals (e.g., mission success, safety, schedule, and budget) while satisfying their specified requirements.
- The NASA Software Engineering Requirements provide a set of software engineering requirements in generic terms to be applied throughout NASA and its contractor community.



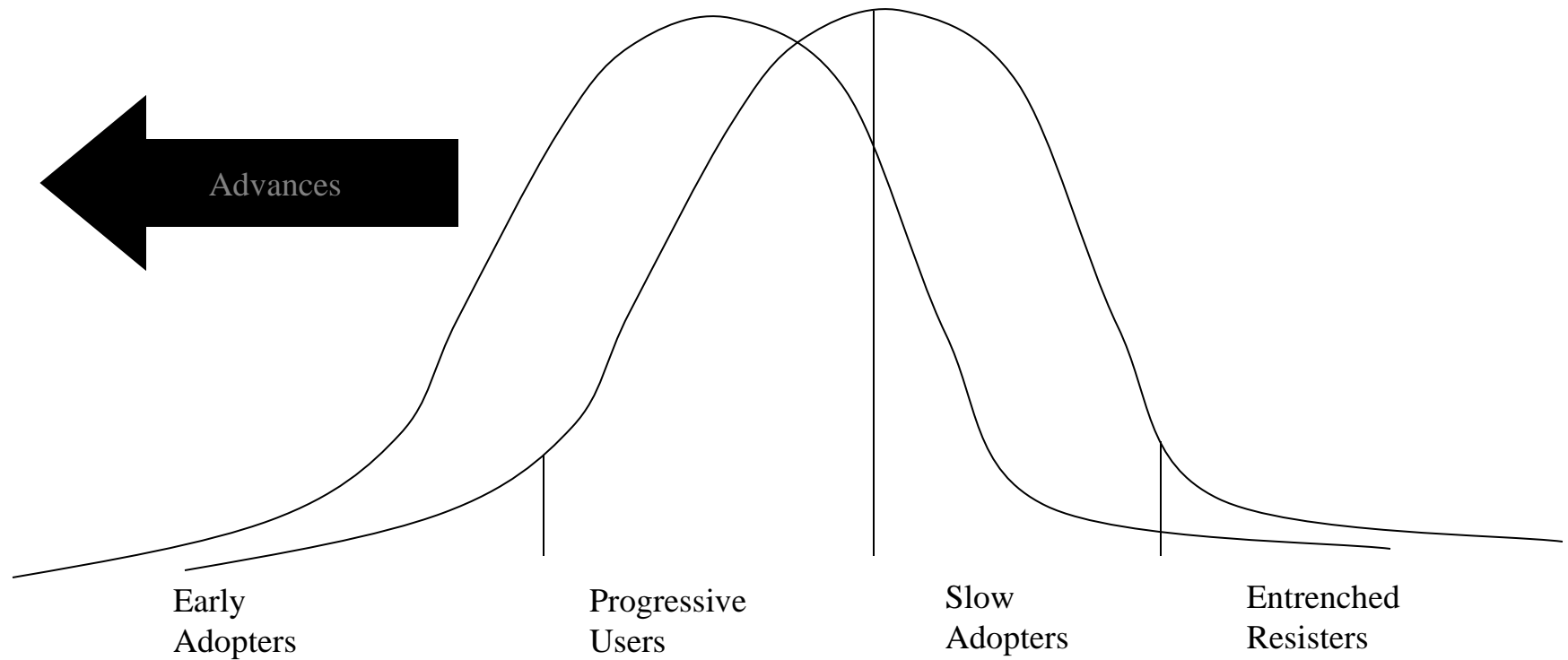
Profile of NPR target audience





Purpose of NPRs

Shift target audience to the left 10 - 25%



This is our target after putting the NPR in place and after each NPR update cycle



NASA-wide Software Classification*

OCE

- Class A** Space Flight Human Rated Software Systems
- Class B** Non-Human Space Rated Software Systems
- Class C** Mission Support Software & Facilities
- Class D** Analysis and Distribution Software
- Class E** Development Support Software

(e.g., Class A – C is mostly software developed or acquired for Highly Specialized IT systems)

CIO

- Class F** General Purpose Computing Software
(Multi-Center or Multi-Program/Project)
- Class G** General Purpose Computing Software
(Single Center or Project)
- Class H** General Purpose Desktop Software

* Established by NPR 7150.2A



NASA Software Engineering Requirements, NPR 7150.2 **CMMI Requirement**

- **[SWE-032] The project shall ensure that software is acquired, developed and maintained by an organization with a non-expired Capability Maturity Model Integration® for Development (CMMI-DEV) rating as measured by a Software Engineering Institute (SEI) authorized or certified lead appraiser as follows:**
 - **For Class A software: CMMI-DEV Maturity Level 3 Rating or higher for software, or CMMI-DEV Capability Level 3 Rating or higher in all CMMI-DEV Maturity Level 2 and Maturity Level 3 process areas for software.**
 - **For Class B software: CMMI-DEV Maturity Level 2 Rating or higher for software, or CMMI-DEV Capability Level 2 Rating or higher for software in the following process areas:**
 - **a. Requirements Management.**
 - **b. Configuration Management.**
 - **c. Process and Product Quality Assurance.**
 - **d. Measurement and Analysis.**
 - **e. Project Planning.**
 - **f. Project Monitoring and Control.**
 - **g. Supplier Agreement Management (if applicable).**
 - **For Class C software: The required CMMI-DEV Maturity Level for Class C software will be defined per Center or project requirements.**



CMMI Requirement Notes

- Note: **Organizations who have completed Standard CMMI® Appraisal Method for Process Improvement (SCAMPISM) Class A appraisals against the CMMI-DEV Model are to maintain their rating and have their results posted on the SEI web site so that NASA can assess the current maturity/capability rating.** Software development organizations need to be reappraised and keep an active appraisal rating posted on the SEI web site during the time that they are responsible for the development and maintenance of the software.
- Note: **For Class A software development only, a transition period to obtain a CMMI-DEV Maturity/Capability Level 3 Rating will be allowed for organizations developing Class A software per the NASA Headquarters Office of the Chief Engineer approved Center Software Engineering Improvement Plan as described in SWE-003, SWE-004, and SWE-108.**
- Note: **For Class B software, in lieu of a CMMI rating by a development organization, the project will conduct an evaluation, performed by a qualified evaluator selected by the Center Engineering Technical Authority, of the seven process areas listed in SWE-032 and mitigate any risk, if deficient. This exception is intended to be used in those cases in which NASA wishes to purchase a product from the "best of class provider", but the best of class provider does not have the required CMMI rating. When this exception is exercised, the Center Engineering Technical Authority should be notified.**



CMM/CMMI Lessons Learned by NASA

- **Preparing for an appraisal is where you get the measurable process improvement**
- **CMMI process helped Centers establish a baseline of where they are**
- **Develop an extensive set of “tools” (i.e., templates, spreadsheets) to help projects with CMMI practices and artifacts**
 - **Use of toolset helped projects reach compliance much faster**
- **Mentors can help get Project tool use started and help Projects tailor the artifacts**
- **Established sponsorship across departments**
 - **Management Steering Group**
 - **Was difficult to get mid-level managers to “own” improvement program**
- **Established early on a relationship with the Lead Appraiser**
- **PIID development and artifact collection**
 - **PIIDs and artifacts were maintained on a server for ease of access and review**
- **Importance of interview preparation and training**



The perfectly picturesque spiral galaxy known as Messier 81, or M81



CMM/CMMI Lessons Learned by NASA

- Use the workshops to review the processes in depth and reinforced the tool sets
- Tracking Progress, determine a method for projects to report progress
- Many of our projects need basic project management and configuration management training
- CMMI assessments helped identify areas for process and project improvement
- Despite initial reluctance, pre-appraisal was a positive experience for our projects - laid a good foundation for future involvement
- Projects appreciated systematic and analytical feedback on what they are doing
- Measurement and analysis is a big challenge
- Improved quality and review of management plan early in the life cycle and reuse of the plans for new projects
- Resource planning and tracking at the individual process level provided little additional benefit to the projects
- Smaller projects need to have light-weight processes to avoid being smothered (especially for a one person task)



Mars
Image Credit: NASA/JPL-Caltech



CMMI Key Impacts at NASA



Space Shuttle Launch picture from Disney's Castle

- **Reduces risk of software failure - Increases mission safety**
 - Improvement processes based on best practices in Industry and Government
 - Risk management much improved on software subsystems--Previously there was little monitoring of risks
- **More predictable software cost estimates and delivery schedules**
 - Data showed projects working within CMMI software framework & best practices had increased accuracy in cost estimates and smaller growth in resources over the lifecycle
- **Smarter buyer of contracted out software**
 - Educating the NASA workforce on best practices in Software Engineering
- **More defects found and removed earlier**
- **Reduces duplication of efforts between projects**
- **Increases ability to meet the challenges of evolving software technology**
- **Software development planning has been improved across the Agency**
 - There is a growing consensus among the practitioners and software managers that working to a defined process has substantial benefits.
 - Vast improvement in planning of software projects and in monitoring progress
- **NASA's contractor community has heard the word that the bar has been raised with respect to software engineering and is responding appropriately**
 - NPR 7150.2A, Software Engineering Requirements (update Nov 2009)



CMMI Key Impacts at NASA

- **A solid foundation and structure is now in place for developing software in a disciplined manner**
 - More uniformity in management plans, reviews, test plans, status reporting Risk management much improved on software subsystems--Previously there was little monitoring of risks
 - Data management and configuration management has improved
 - Improve the working relationships between Engineering and Safety and Mission Assurance with respect to software engineering
- **The Agency is better prepared for major programs and projects than it was 8 years ago**
 - Software teams and software quality engineers are working together to assure compliance to standards, to improve quality
- **The knowledge and skills of the NASA software engineering community has significantly improved**
 - We have seen significant cultural changes.
- **Extensive mentoring program established to improve software practices**
- **Our projects are now better managed – particularly in the area of progress tracking**
 - Now we know exactly where we are in the project and how long it's likely to take to finish



Space Shuttle Atlantis Rollout at Kennedy Space Center April 2010



Summary of NASA's Experience with CMM and CMMI

Why improve processes? Because process is the foundation for all other improvements, and lasting improvements are not possible without it.

If a performance management system is not in use, leadership is unaware of what is and is not working.

CMMI is a proven approach to performance management – with more than a decade of results showing it does work.

Simply deciding to “do CMMI” is not enough to achieve benefits.

Defining good processes, using them, measuring the results, and making improvements based on what you learn are all key to reaping the benefits of process improvement.

The CMMI models are one part of a comprehensive approach to process improvement that helps organizations understand

- *why* they should improve
- *what* frameworks and tools would best fit their needs
- *how* to implement them



Apollo 8, the first manned mission to the moon