AMES SUPERCOMPUTER: This Cray 2 Supercomputer for NASA's numerical aerodynamic simulator is found at the Ames Research Center in Mountain View, CA.

Photo by Wade Sisler, NASA

DOCUMENTATION

Developing Best In Class Processes at NASA

by **Tim Olson and John C. Kelly**

Almost everyone in the United States is familiar with NASA and its well-publicized space activities. But NASA also performs many less well-known engineering activities such as developing processes.

In 50 Words Or Less

- **For the last few years, NASA has successfully used best practices to develop short, usable processes and procedures.**
- **Processes should have different levels of documentation—beginner, intermediate and expert.**
- **A good process describes who, what, where, when, why and how and uses pictures.**

Systems engineering and software engineering are core capabilities and key enabling technologies necessary for the support of NASA's enterprises. Ensuring the quality, safety and reliability of NASA systems is of paramount importance in achieving mission success.

Many system challenges within the agency have been identified and documented through surveys and assessments. The exponential growth in the scope, complexity and importance of systems within NASA over the years is expected to continue, challenging NASA's ability to manage them effectively.

As a result, NASA instituted a software engineering initiative and similar activities for systems engineering under the responsibility of the office of the chief engineer.[1] Within the project planning, requirements and design initiatives, best practices have been developed for processes. Table 1 shows common problems related to process documentation.

How can you address the common problems with process documentation? One place to start is to recognize not all documentation is used the same

TABLE 1 Common Process Documentation Problems

Problem	Description
1. Too big	Most process documentation is too big. Blaise Pascal once said, "I have made this letter longer than usual because I lack the time to make it shorter."* This quote applies to most processes and procedures. Process documentation should be short, concise and usable.
2. Not enough pictures	Most processes and procedures lack pictures and diagrams. If a picture is worth a thousand words, process documentation should have more pictures. Good process documentation should be a mixture of pictures and words. The best pictures are well-thought-out diagrams. The best diagrams for process documentation are process models.
3. Poorly designed documentation	Processes and procedures usually violate documentation design and good writing principles. Principles such as chunking and consistency are usually not followed.*
4. Unusable and one size fits all	Most processes and procedures are not designed with customers and users in mind, making them hard to use. Much documentation also has the one size fits all mentality because it does not consider expert, intermediate and beginner users.
5. Mixed information types	Policies, standards, processes, procedures and training are all different types of information.** Most process documentation mixes these different types of information into the same paragraphs as if they were all used the same way. Each one of these document types has a different usage scenario.
6. Written sequentially	Process documentation is not a novel and is not meant to be read linearly (from beginning to end). Process documentation is reference material meant to be used nonlinearly. Labeling is critical because it allows users to find information quickly.*
7. Information that can't be found quickly	Users of documentation will look for information for a few minutes. But if they cannot find the information quickly, many times they will give up in frustration and not use the processes or procedures. This can lead to serious nonconformance problems for many organizations.
8. Documentation becomes shelfware	Most process documentation becomes shelfware, collecting dust. Online processes must be well designed or they will also become unused webware.

* Robert E. Horn, "Developing Procedures, Policies and Documentation," Information Mapping Inc., 1992.

** Timothy G. Olson, Neal R. Reizer and James W. Over,) "A Software Process Framework for the SEI Capability Maturity Model," CMU/SEI-94-HB-01, 1994.

Table source: Timothy G. Olson, "How To Define Short and Usable Processes," paper and presentation, 13th ASQ International Conference on Software Quality, Dallas, 2003.

TABLE 2 Types of Process Documentation

Document type	Usage
Policy	• Used by senior management to set direction in an organization. • States principles that organizations should follow.
Standard	• Specifies the parts of a document and provides a description of what goes into those parts. • Makes the content of documents repeatable.
Process	• Tells what happens over time to produce desired results. • Should answer the 5 W's (who, what, where, when and why) and how.
Procedure	• Provides how-to or step-by-step information.* • Implements part of a process.

* Robert E. Horn, "Developing Procedures, Policies and Documentation," Information Mapping Inc., 1992.

Table source: Timothy G. Olson, "How To Define Short and Usable Processes," paper and presentation, 13th ASQ International Conference on Software Quality, Dallas, 2003.

way; "process documentation" can refer to policies, standards, processes and procedures. Table 2 shows these types of process documents and how each is used in practice. The documentation framework of Figure 1 (p. 60) identifies the types of process documents and some critical relationships among them.

Usage Modes

Processes and procedures have different levels of users, each requiring a different level of documentation:[2]

- Experts have used the process many times and may even be responsible for running the process.
- Intermediates have used the process a few times but need guidance and lessons learned.
- Beginners have never used the process.

Expert. Expert mode documentation is short, concise and does not contain any training material.[3] When a pilot flies an airplane, he or she does not pull out training manuals. Instead, pilots use expert checklists for takeoff and landing.

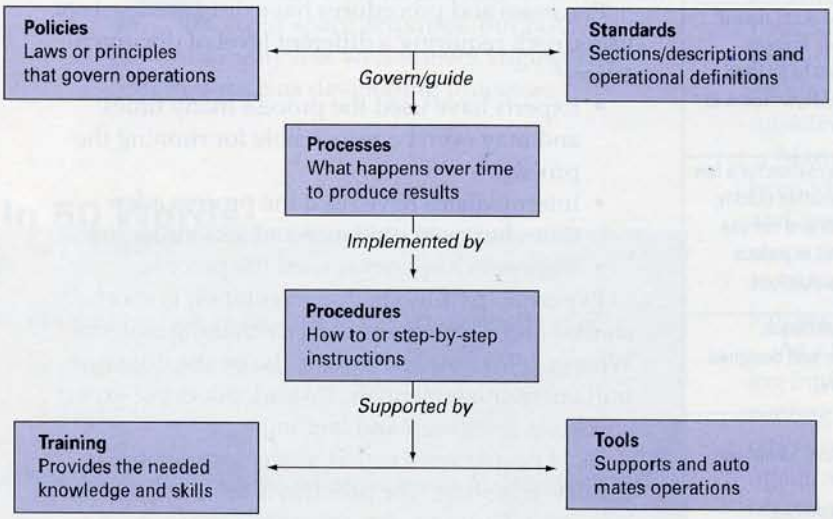
Most people want expert mode documentation because it is short. The problem is not everybody is an expert. For example, not everyone can read a checklist for a rocket scientist—sometimes you really need to be a rocket scientist!. Putting expert mode documentation into the hands of nonexperts can be dangerous.

Why do experts need documentation if they are experts? People can forget things. This is why checklists are so powerful. Expert knowledge should be documented because experts can leave your organization, taking precious organizational knowledge with them.

Intermediate. Intermediate mode documentation uses the expert mode documentation but builds and adds to it by providing guidance and lessons learned. Guidance is very useful to people who don't have to follow a process or procedure very often. Even experts forget guidance and lessons learned for an annual or infrequently used process. Having guidance available to those who need or want it is very useful.

Typically guidance and lessons learned are not auditable. Process phases and procedure steps are required and auditable, but the supporting guidance and lessons learned are there for support only. One best practice is to distinguish between required steps and optional guidance. NASA has chosen to mark guidance and lessons learned with a "guidance" label. The NASA ISO 9000 group prefers this approach for auditing.

FIGURE 1 Documentation Framework



Adapted from "A Software Process Framework for the SEI Capability Maturity Model" Timothy G. Olson, (Quality Improvement Consultants Inc, 2005).

Beginner. Beginner mode documentation uses the intermediate mode documentation, but adds training. Beginners should feel free to use the training manuals until they become familiar with the process. Beginners should also be mentored as appropriate. Some processes are simple, and some are complex. Complex processes should have formal training followed up by mentoring.

How can an organization afford to provide three versions of the same documentation? Someday software might include a documentation mode (expert/intermediate/beginner) allowing the user to see the appropriate information.

Until that time, a best practice that solves this problem develops the process in chunks (sections) in one version at the intermediate level. Add training for the beginner, and let the expert grab the appropriate chunks. Another best practice is to provide expert mode documentation for the experts.

Best Practices

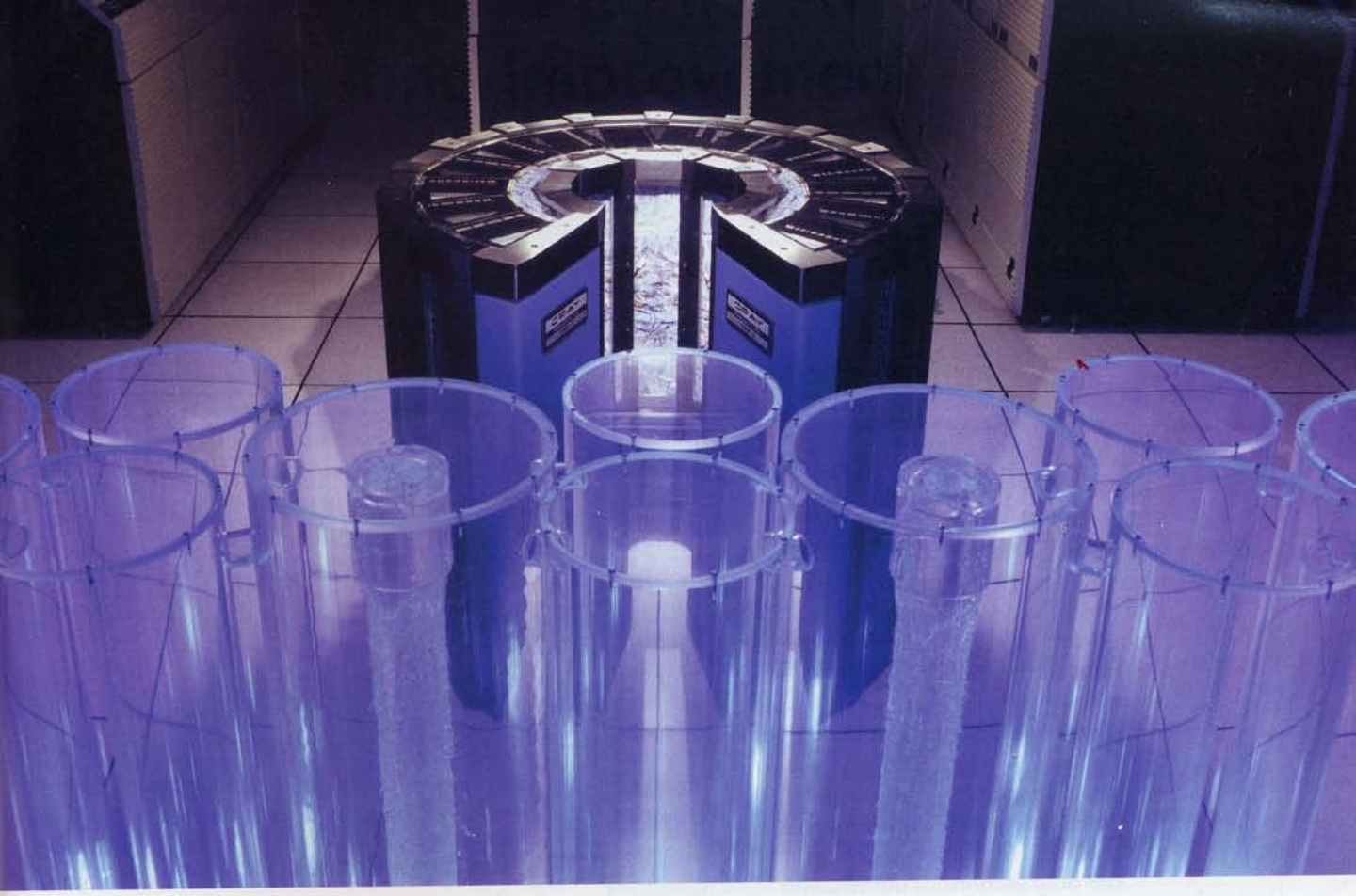
Process documentation works well only if the process it describes contains all the necessary elements. Table 3 describes the process elements—the who, what, where, when, why (five W's) plus how—required to develop a good process.[4] Each question is answered by a key process element.

A "good process" should:

- Address the five W's and how, and answer the key process questions in Table 3.
- Have both pictures and words (most people prefer pictures, but some people prefer words).
- Be short, usable and well written.
- Be well sectioned and labeled (so chunks can be found quickly).[5]

Although a good process should have pictures, not all pictures are good ones. Some pictures cause confusion, and some are more harmful than helpful. So what is a good picture? Process modeling is a best practice that helps design good diagrams that address the five W's.

Figure 2 (p. 62) is an example



UP CLOSE: A view of NASA's 10,240 processor SGI Altix supercomputer housed at the advanced supercomputing facility.
Photo by Tom Trower, NASA Ames Research Center

of a process model. A process model answers questions about reality[6] and is typically represented by diagrams and powerful notations that show roles, activities, work products and the relationships among them.

Best in class procedures consist of how-to, step-by-step information and come in three formats:[7]

- Checklists.
- Forms.
- Step/action tables.

Checklists. Checklists are powerful, repeatable representations of activities needed to complete something. A checklist is powerful because the order in which it is completed usually doesn't matter. This makes checklists useful for concurrent activities, as opposed to flowcharts, which are very poor at representing concurrency.

Forms. Forms, along with instructions for completing them, are repeatable mechanisms for supporting processes. Forms are powerful mechanisms for collecting data in a repeatable way.

Step/action tables. One effective way to represent a procedure is to use a step/action table,[8] which is useful when order matters. For example, if a person needs to track his or her time, starting to track time should not be the last step. Table 4 (p. 62) is an example of a step/action table.

TABLE 3 Key Process Questions

Key process question	Process element
Why is the activity performed?	Purpose
Who does what activity?	Activity performed by roles
What work products are used?	Inputs
What work products are produced?	Outputs
When does the activity begin	Entry criteria
When does the activity end?	Exit criteria
How is the activity implemented?	Subactivity, procedure, method
What activity is next?	Flow (such as, sequence or selection)
Where is the activity performed?	Context (such as hierarchy)

Adapted from "A Software Process Framework for the SEI Capability Maturity Model" (Timothy G. Olson, Quality Improvement Consultants Inc, 2005).

NASA Success Stories

NASA designs and builds complex systems. This requires complex processes and procedures. Several NASA locations have used best practices in process documentation successfully recently. Figure 2 (p. 62) is an example of one of NASA's simpler processes.

NASA's engineering process groups (EPGs)

develop, measure and improve systems. Some of these groups became frustrated when their process documentation grew large, complex and difficult to use and therefore decided to try some process definition best practices.

Langley. The first group to pilot these process definition best practices was the EPG at NASA's Langley Research Center in Hampton, VA. The first procedure to be improved was the software process for the center. Although the old procedure was above average from a process benchmarking point of view, it had a few weaknesses:

- **There was a mixture of document types:** Policies, standards, processes and procedures were mixed together.
- **The principle of chunking was violated.** Most people can remember only sections of information (for example, that is why 15- to 16-digit credit cards numbers are broken into smaller chunks of four or five numbers). The original NASA Langley software process had a flowchart with more than 30 steps and a procedure with more than 25 steps, making it difficult to use.
- **Processes did not address all the five W's and how.** "When" was missing, and other W's were weak.

The new procedure implemented best practices and addressed all these weaknesses. Process modeling was used to add good diagrams and address chunking of the flowchart. The new processes addressed all the five W's and how and were developed on one page for the expert mode. NASA Langley has now developed about six new and improved NASA procedures based on best practices.

NASA Engineering and Safety Council. The recently formed NASA Engineering and Safety Council (NESC) developed about 30 procedures. Although NESC had well above average procedures from a benchmarking perspective, the agency wanted them to be shorter and more usable. Using best practices, NESC cut the procedures by 60%

TABLE 4 Step/Action Example

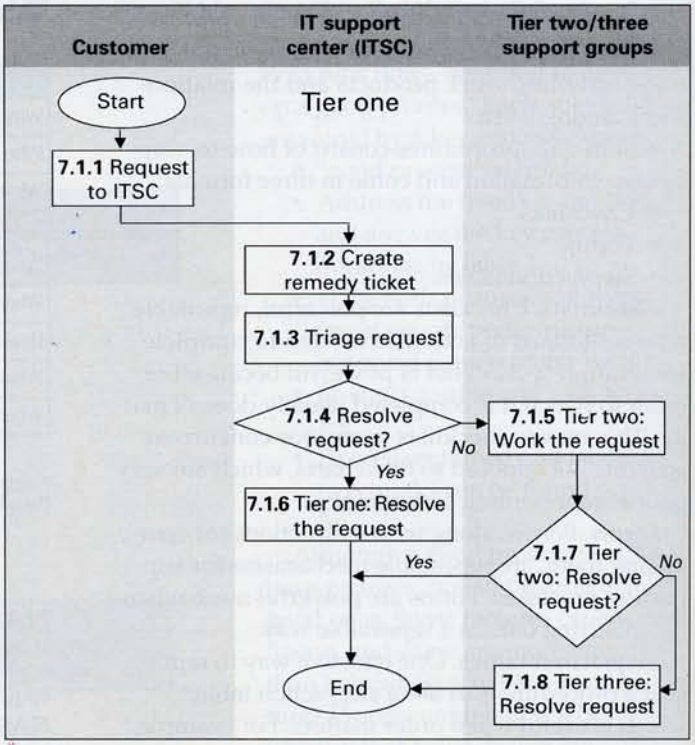
	Step
1	Begin to track time (write down the start time).
2	Look for defects in the selected work product by using the appropriate data driven checklist.
3	Log the defects on the defect form. Continue logging defects until the work product is completely inspected using the checklist.
4	End tracking time (write down the end time). Calculate the total time spent looking for and logging defects, and record the total time on the defect form.

Table source: Timothy G. Olson, "How To Define Short and Usable Processes," paper and presentation, 13th ASQ International Conference on Software Quality, Dallas, 2003.

FIGURE 2 Customer Request Process

Purpose: To address customer requests effectively and efficiently.

Inputs/entry criteria:
- Customer request received (phone call, voice mail, e-mail, fax, interoffice mail, walk through).
- Remedy ticket created.



Outputs/exit criteria:
- Customer request is resolved.
- Request is closed in remedy.

without affecting technical content. Now the NESC procedures are not only shorter; they are more usable.

Ames IT Support Center. The NASA Ames Center is located in the Silicon Valley in Mountain View, CA. Its IT Support Center (ITSC) had an experienced manager who had built a customer support department from the ground up. Although detailed procedures existed for addressing trouble tickets, the overall customer support process had not been documented.

NASA Ames ITSC developed a new document with four processes in 19 pages (similar processes run hundreds of pages). The process was developed in intermediate mode and also used process modeling for pictures. The five W's were put into a one-page diagram for each process in expert mode, and a page of text (along with guidance) was developed in intermediate mode to support the diagram. Table 5 and Figure 2 are from NASA Ames.

The experienced manager at NASA Ames was recently promoted and is now working in another area of NASA. The new process document played a key part in her promotion because it allowed a new manager to come in and learn the ITSC process quickly. Sometimes experienced people get stuck in positions because they are the only ones who know the process, and the process is not documented.

Code M at NASA Headquarters. NASA headquarters is located in downtown Washington, DC. Historically, its departments have been named with codes using the alphabet. Code M, which developed and maintains the NASA shuttle and the space station, is the largest.

The chief engineer of Code M wanted a process review and benchmarking of all its procedures at headquarters. The process review identified strengths and opportunities for improvement and made specific recommendations to Code M senior management.

During the process review, it was discovered a key process was not documented. This process was developed with the chief engineer of Code M, and the new process is 19 pages long. The process was developed in intermediate mode and also used process modeling for pictures, similar to Figure 2. The five W's were put into a one-page diagram in expert mode, and a page of text (along with guidance) was developed in intermediate

TABLE 5 Customer Request Process

Step number	Process step
7.1.1	**Request to IT support center (ITSC)** • Customer requests a service or reports a problem via phone, voice mail, e-mail, fax, interoffice mail or walk through. Guidance: A customer can contact the ITSC during normal business hours by using the following options: • Call x4-2000 or fax x4-0063. • E-mail help@mail.arc.nasa.gov. • Go to building 233, room 195. • Send interoffice mail to mail stop 233-17.
7.1.2	**Tier one: Create remedy ticket** • ITSC staff will use the remedy help desk application to creat a remedy ticket for all customer requests. • Staff should capture clear, correct and complete information into the remedy ticket and within the metric stated in the service level agreement (SLA).
7.1.3	**Tier one: Triage request** • ITSC staff will need to follow the SLA and standard operating procedure (SOP) to correctly determine the assignment of a remedy ticket. Guidance: Triage is referencing the SLA and SOP and using professional judgment to resolve the request or correctly assign the request.
7.1.4	**Tier one: Resolve request?** • If yes (you can resolve the request), then proceed to process step 7.1.6. • If no, then assign the request to tier two. Guidance: • Once a request is assigned to tier two, it is ITSC's responsibility to ensure it is addressed. • The ITSC lead will check tier two and tier three tickets daily for ones that have remained in the new status for over eight hours and send a report of those tickets to tier two and tier three managers.
7.1.5	**Tier two: Work request** • Follow the SLA to resolve the request. • Update the remedy worklog with clear, correct and complete information. • Change the remedy status to "work in progress" or "pending."
7.1.6	**Tier one: Resolve request** • Follow the SLA and SOP to resolve the request. • Update the remedy worklog with clear, correct and complete information. • Change the remedy status to resolved. Guidance: Ensure customer is satisfied with resolution.
7.1.7	**Tier two: Resolve request?** • If yes (you can resolve the request), then change the remedy status to resolved. • If no, then assign the request to tier three. Guidance: Ensure customer is satisfied with resolution.
7.1.8	**Tier three: Resolve request** • Follow the SLA to resolve the request. • Update the remedy worklog with clear, correct and complete information. • Change the remedy status to resolved. Guidance: Ensure customer is satisfied with resolution.

mode to support the diagram. The new process also met the ISO 9001 requirements for NASA.

Office of the Chief Engineer. The office of the chief engineer, located at NASA headquarters, is periodically charged with writing or updating documents and requirements for all of NASA. The program executive for software engineering was charged with writing the agency level policies and procedural requirements for software developed, maintained or acquired for NASA.

The new procedural requirements document was developed in about 50 pages with about 140 requirements. This is much shorter than most other similar industry standards, which are hundreds of pages long.

Requirements used the word "shall" and were written in one sentence. Optional recommendations were added as notes. The requirements were chunked into software best practices, and a short description introduced each best practice.

Eight classes of software based on the type of subsystem and how critical the software is to project success were identified at NASA. The 140 requirements were tailored to each class using a matrix. The human rated mission critical software had all 140 requirements apply, but fewer requirements were applied to lower classes and different types of NASA software.

Some Lessons Learned

Some of the lessons learned at NASA and many other organizations in industry while developing processes with best practices are:

- Don't mix policy, standard, process and procedure information (for example, in the same paragraph). Label this different information, and consider how the information is used.
- Write all process documentation as simply as possible but not so simply it doesn't work. Keep process documentation concise and precise (short and sweet), but expect some processes to be complex.
- Use good pictures (most people prefer pictures), with words filling in the details.
- For each process or subprocess, write the five W's and how on one page using a diagram, as in Figure 2. A good process diagram can replace 20 to 25 pages of text.
- Process modeling is a best practice and scales up to extremely complex systems. Use process modeling to develop good pictures.
- Information Mapping is also a best practice.[9]
- Use procedures such as checklists, forms and step/action tables.[10]
- Use chunking or sectioning to organize the sections, and label the chunks so users can find information quickly. Process modeling and Information Mapping help tremendously with this principle.
- Account for beginner, intermediate and expert users of the process.
- Design measurement into the process. Don't add measurement as an afterthought.
- The processes must be tailored to each organization, each business unit or division and each project.

Developing best in class processes and procedures—making them short and usable—is challenging. Many best practices can be used to help improve process documentation. The approach summarized in this article uses a collection of best practices, all wrapped into a process for developing processes.

Although this process of developing processes is outside the scope of this article, we hope readers have benefited from the description of some of the best practices along with the example process in Figure 2 and Table 5.

ACKNOWLEDGMENT

Pat Patterson, manager of the NASA engineering and training (NET) program, recognized a common problem with training is that little or nothing changes when people go back to their jobs. The strategy used at NASA was to follow the process definition training with hands-on workshops to actually implement the training. The NASA success stories described in this article are a direct result of the funding of Patterson's NET program.

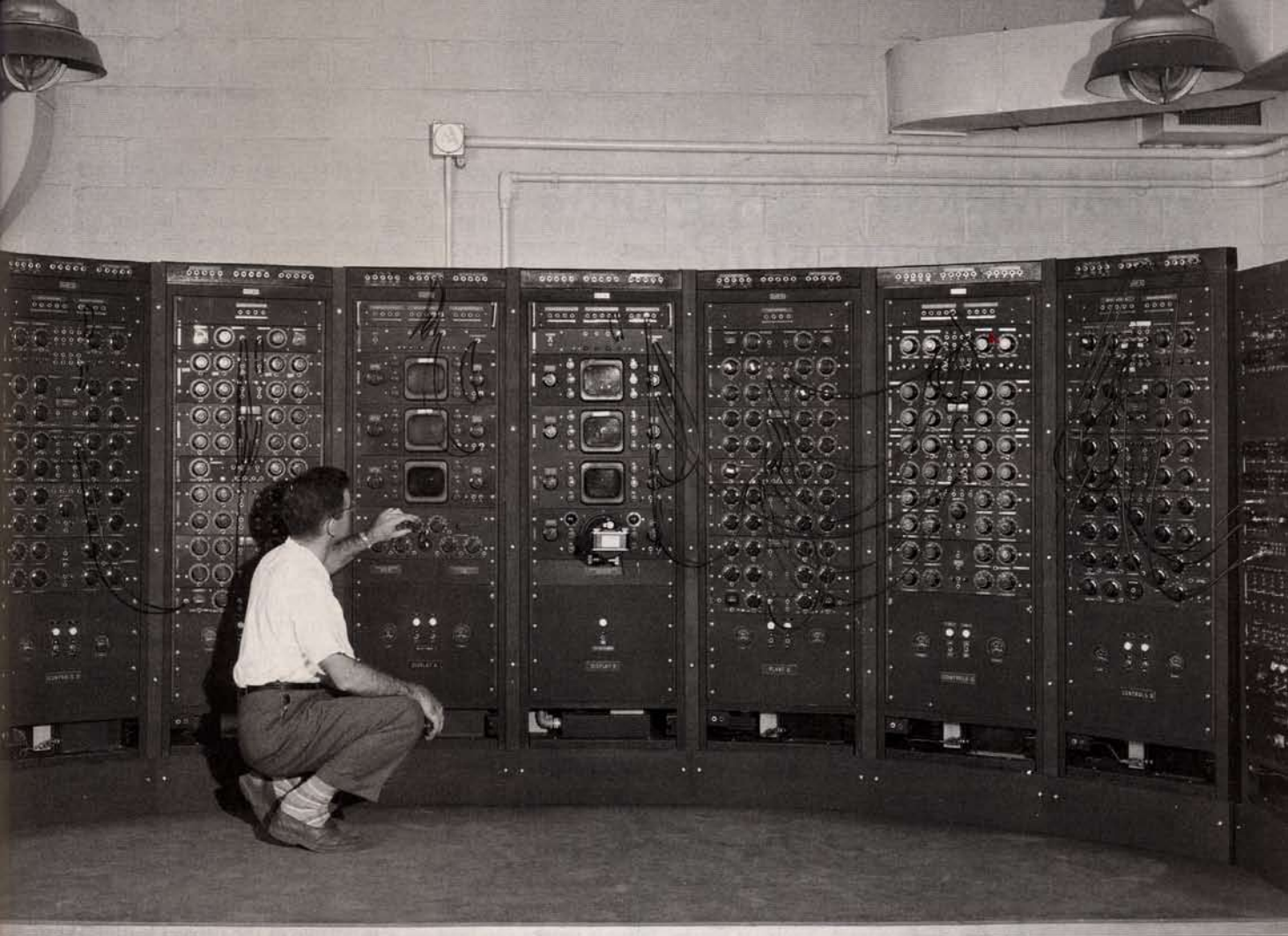
REFERENCES

1. "NASA Software Engineering Initiative Implementation Plan," NASA, 2002.

2. Timothy G. Olson, "Defining Software Processes in Expert Mode," presentation, NASA Software Engineering Process Group Conference, Atlanta, 1999.

3. Ibid.

4. Timothy G. Olson, Neal R. Reizer and James W. Over, "A Software Process Framework for the SEI Capability



THE OLD DAYS: This analog computing machine, an early version of the modern computer, was housed in NASA's fuel systems building in 1949. Today it's at the John H. Glenn Research Center in Cleveland.

Photo by Fred Lingelbach, NASA

Maturity Model," CMU/SEI-94-HB-01, 1994.

5. Robert E. Horn, "Developing Procedures, Policies and Documentation," Information Mapping Inc., 1992.

6. Douglas T. Ross and Kenneth E. Schoman Jr., "Structured Analysis for Requirements Definition," *IEEE Transactions on Software Engineering*, SE-3(1), 1977, pp. 6-15.

7. Olson, "Defining Software Processes in Expert Mode," see reference 2.

8. Horn, "Developing Procedures, Policies and Documentation, see reference 5.

9. Ibid.

10. Ibid.

TIMOTHY G. OLSON *is founder and president of Quality Improvement Consultants Inc., San Diego, CA, and an associate with the Juran Institute, Southbury, CT. He has been formally trained in Crosby, Deming, Juran, ISO 9000, capability maturity model and capability maturity model integration quality approaches. Olson is a lead-author of a software quality course for the University of Minnesota and is a member of ASQ. He holds a master's degree in computer science from the University of Massachusetts, Lowell.*

JOHN C. KELLY *is the program executive for software engineering within the NASA headquarters office of the chief engineer. He also serves as chair of the NASA software working group and is a member of the capability maturity model integration steering group. He earned a doctorate in mathematics/education from Florida State University.*

Please comment

If you would like to comment on this article, please post your remarks on the *Quality Progress* Discussion Board at www.asq.org, or e-mail them to editor@asq.org.