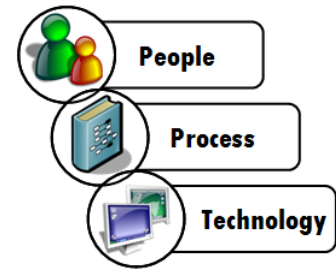


Technology Capability Documentation Operating Guidelines

Version 0.4, Updated: 5/31/2021



Contents

Introduction	1
1. "Just Enough Process".....	1
2. KISS Documentation	1
3. "Just Barely Good Enough" Documentation	2
4. "80/20" Everywhere	2
5. A Bias toward Buying versus Building	2
Standard for Standards	3
1. Quality	3
2. Productivity	3
3. Fit for Use	3
Appendices	4
1. Tips for Defining a Process.....	4
2. Tips for Creating Diagrams and Graphics	5
3. General Design Tips for Shared Templates	5

Introduction

This document describes the Operating Guidelines used to build technology-based capabilities. It is based on industry best practices and it provides guiding principles for effective process (and, by extension, capability) management. A summary of 4 of these practices is provided in this Introduction section.

1. "Just Enough Process"

For effective and maintainable process management, GartnerGroup advises a strategic balance between two extremes:

No documented processes	All processes documented
Anarchy	Bureaucracy and red-tape
Flexibility	Discipline
Creativity and innovation	Scalability and repeatability

2. KISS Documentation

Extracted from: http://www.infoworld.com/article/04/02/27/09OPconnection_1.html and <http://www.agilemodeling.com/essays/agileDocumentation.htm>

In this context, *KISS* stands for "Keep It Short and Simple." From the above-referenced web sites are these guidelines: "The best documentation is the simplest that gets the job done. Don't create a 50-page document when a five-page one will do. Don't create a five-page document when five bullet points will do. Don't create an elaborate and intricately detailed diagram when a sketch will do. Don't repeat information found elsewhere when a reference will do. Write in point form. Document only enough to provide a useful context."

Critical Points	Increasing the Agility of Documentation
• The fundamental issue is communication, not documentation. • Documentation should be "lean and mean". • Travel as light as you possibly can. • The benefit of having documentation must be greater than the cost of creating and maintaining it. • Each system has its own unique documentation	• Focus on the Consumer • Keep it just simple enough, but not too simple • Require people to justify documentation requests • Write the fewest documents with the least overlap • Get someone with proven writing experience

needs; one size does not fit all. • Ask whether you NEED the documentation, not whether you want it. • The investment in system documentation is a business decision, not a technical one.	
--	--

One helpful exercise is to see if a document can be described, in total, on a single page. If so, that's all you need. If not, how about a 2-sided reference card? This exercise forces an economy of words.

3. "Just Barely Good Enough" Documentation

Extracted from: <http://www.ddi.com/architect/184415826?cid=Ambysoft>

"Just Barely Good Enough" Documentation:

- Maximizes stakeholder investment. A document's benefit is greater than the investment in its total cost of ownership (TCO), and ideally the investment made was the best option available for those resources.
- Is lean. Contains just enough information to fulfill its purpose and no more. Follow the DRY (Don't Repeat Yourself) principle.
- Fulfills a single purpose.
- Isolates into one place the information that is likely to change quickly.
- Describes "good things to know." KISS documents capture critical information, information that isn't obvious, such as design rationale, requirements, usage procedures and operational procedures.
- Has a specific consumer and facilitates the work efforts of that consumer. Different consumers require different types of documents, and very likely different writing styles. You must work closely with the consumer, or potential consumer, for your documentation if you want to create something that will actually meet their needs.
- Is sufficiently accurate, consistent and detailed.

4. "80/20" Everywhere

Extracted from: http://en.wikipedia.org/wiki/Pareto_principle

From Wikipedia is this definition: "The Pareto principle (also known as the **80-20 rule**, or the **law of the vital few**) states that, for many events, 80% of the effects comes from 20% of the causes...In business, dramatic improvements can often be achieved by identifying the 20% of customers, activities, products or processes that account for the 80% of contribution to profit."

Coupled with the above principles, the 80/20 rule can be applied by the IS Capability team in a number of ways:

- Portfolio Management
 - Given the number of efforts that we could potentially address, we need to make sure we apply appropriate criteria and work on only the "vital few" projects. We need the organizational maturity to be able to "just say no" (for now) to initiatives of lesser importance.
 - This is in concert with the "Just enough process" principle stated above.
- Project Management
 - Once a project is selected from our portfolio, and the project is begun, again we need to apply the 80/20 rule and identify if there is a subset of the full-blown project that might deliver the bulk of the desired benefit.
 - This is in concert with the principles of "KISS" and "Just Barely Good Enough" principles.
- Elaborating Alternative Solutions
 - Sometimes, through innovative or out-of-the-box thinking we may be able to identify even smaller subsets of work than the 80-20. Some of these might be considered "quick and dirty" solutions; others might be "low-hanging fruit."

The Wikipedia article goes on to state: "If the parameters in the Pareto distribution are suitably chosen, then one would have not only 80% of effects coming from 20% of causes, but also 80% of that top 80% of effects coming from 20% of that top 20% of causes, and so on (80% of 80% is 64%; 20% of 20% is 4%, so this implies a "64-4 law)."

It is by rigorous application of these principles that we can maximize our contribution to the organization. Despite the large number of organizational needs, we may be able to achieve 64% of the benefit with a mere 4% of the effort (staff) that doing all of it would take.

5. A Bias toward Buying versus Building

Whenever possible, we will prefer to buy what we need rather than build it, such as for methodologies, assessments, etc.

Standard for Standards

1. Quality

Written processes and standards are expected to be above reproach; typos are unacceptable. Hence, we must ensure that:

1. New processes (and process changes) are implemented right the first time
 - Process development teams must include the SMEs from the start
 - Process development teams must involve reps from all impacted areas, if not on the team itself, at least in the review steps
 - Process development teams must employ easy-to-use, standard templates
 - Automation of standards and procedures should be done to the degree possible
 - Mechanisms to measure compliance and effectiveness of the process must be built in
 - Process documentation is easily accessible and available when needed
 - Process changes are properly announced with sufficient advanced lead time
 - Any training and support needed to effectively implement the process is delivered on time
 - The pacing of process change implementations is centrally managed to preclude too much change too quickly for any given audience
2. Processes must be kept current
 - We must develop processes that are as maintenance-free as possible, and have the organizational maturity to update them when needed
 - Some processes may be given a “review” date as well as an “effective date” – at that time, the process may be considered no longer binding, unless it has been reviewed and updated
 - Simple updates to keep data current will follow a different process than enhancements designed to improve a process’s performance
3. Effective processes include ongoing measurement with and are maintained as needed
 - Processes cannot be announced and forgotten – despite the best plans, most processes will need adjustment after they are first implemented
 - The 1.0 version of a process is normally considered a “pilot” version. Normally, its development is guided by the principle of “progress, not perfection.”
 - We must plan for the cost of “living there” as well as the cost of “getting there”
 - Process development teams will provide ongoing status reports to the IS Capabilities manager
 - The IS Capabilities manager will determine what actions to take based on these reports

2. Productivity

Given that processes must be maintained, we must ensure that:

1. We create simple processes and short documentation (see KISS above)
2. To the degree possible, simple updates should follow a routine procedure, and offloaded to administrative associates; this allows process engineers, content developers, and web designers to focus on process creation and enhancement
3. Because we want to offload maintenance chores, the simplest appropriate tool suitable should always be used. Examples:
 - For websites, use Wix, Weebly, or SquareSpace, instead of WordPress; use of the standard templates (section 6) will ensure a consistent look and feel
 - For simple graphics, PowerPoint is preferred over PhotoShop

3. Fit for Use

Documentation must be easy to use:

1. Processes that are very detailed should be documented hierarchically (i.e. in leveled sets). At the highest level, a context diagram ought to be written.
2. The difference between mandatory processes (policies and standards) and optional ones (guidelines) must be clear in the documentation.

3. Documentation needs, first of all, to communicate, and so must be appropriate to the audience. Only when a specific audience really needs it should tutorial or background materials be provided. In this case, it should be kept separate (e.g. linked to) so the key documentation stays at the forefront. Also, linking to previously-written or publicly-available material is more efficient than writing it from scratch.

Appendices

1. Tips for Defining a Process

First, clarify the context for your process by asking:

- What triggers the process?
- What are the inputs delivered to the process and the outputs expected from it?
- Are there existing, standard templates for each input, output and internal handoff? If not, work to gain agreement on the content of these from the other parties involved; proceeding without these details relatively well-understood may lead to wasted effort.
- Are all handoffs in the process agreed to by the suppliers and receivers?

Identify the 5-7 major steps of your process:

- Document how the inputs you're given are transformed into the outputs you need to produce.
- Use the **"sunny-day scenario."** That is, document only how a single transaction flows through the process *assuming that there are no errors, exceptions, failed approvals, etc.* In practice, this generally means:
 - Decision paths are not shown.
 - Loopbacks in the process flow are not shown.Error and exception processing will be documented later—probably in the supporting narrative—but not in the flow diagram.
- Show only key approval points. In the diagram, assume the transaction is approved.

If any major step requires further breakdown, apply the same rules as above.

- Key: Stay at the highest level you can for as long as you can.
- Make sure the inputs and outputs balance across the levels. For example, if a high-level process X turns an input A into an output B, in the exploded view of process X, A should still be shown as an input, and B as an output.

Guideline on depth of detail:

- Write for a reader who is a "TCP" – a Trained, Competent Professional. You will ensure that your process will be rolled out appropriately, so your audience will be "trained" if needed. And our audience members are professional IT folks, so don't spell out the obvious, e.g. what USCC, IS or SDLC stand for.
- Define acronyms that need it, but do it once in a glossary. Then, consistently use the acronyms rather than spelled out terms. The text will read faster and more accurately this way.
- Ideally, you want your documentation to be easy to assimilate and easy to maintain. Aim for short and simple documentation.

Optionally:

- When doing group brainstorming to develop or negotiate process steps, use a plotter-size printout showing each process participant in its own swim lane. Or, use yellow sticky notes on a white board.
- When ready to document the process, use the standard IS process documentation template. As a proven, comprehensive template, it has these components built into it:
 - An auto-generated table of contents
 - Document introduction (Audience, purpose, etc.)
 - Context diagram
 - Swim Lane Diagram, with supporting components:
 - Process narrative
 - Roles and responsibilities chart (RACI)
 - Deliverables matrix
 - Meetings matrix
 - Metrics definition

2. Tips for Creating Diagrams and Graphics

1. Most graphics embedded on a page will likely be seen more frequently on the screen than in print. Hence:
 - a. Design in landscape mode, not portrait
 - b. Don't use italics, or Serif fonts, such as Times New Roman, except for short bits of large-sized text.
 - c. Non-Serif fonts, such as Arial, read better on computer screens; Verdana is designed for the screen.
2. Use spot color only. Use light colors. Keep text on white or near-white backgrounds.
3. Use square boxes or slightly-rounded boxes for text (circles don't hold much text)
 - a. Keep the text as large as possible.
 - b. Minimize the white space between boxes, to maximize the box size itself.
 - c. Use AutoShape connectors to keep shapes connected during moves.
4. If moving to the web, photos should be in compressed JPG format.

3. General Design Tips for Shared Templates

If you need to create a template for others to use in creating their documents, follow the suggestions detailed here.

General.

5. Design for online viewing rather than print media, since most people will access documents online. This means the following:
 - a. As explained above, avoid italic and Serif fonts. Bolding can also be less readable.
 - b. Again, use white backgrounds.
 - c. Use color only when it's needed. Light Gray headings can be effective:

A	B	C	D	E	F
Data	Data	Data	Data	Data	Data

If you need to use color, use the lightest shades of colors available. For example, repeat Light Gray/Little Yellow/Light Green, etc. rather than using multiple colors of darker intensity.

For example:

A	B	C	D	E	F
---	---	---	---	---	---

is better than:

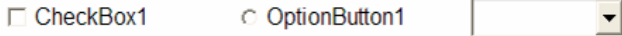
A	B	C	D	E	F
---	---	---	---	---	---

Also, be aware that some folks cannot see the difference between red and green.

6. Design for the web. The above points apply, but further:
 - a. It may be safest to assume that everything you develop may someday be accessed via CellSite.
 - b. Design toward the 4X3 dimensions of the typical user's screen or monitor. This means 1-page diagrams would be laid out as landscape instead of portrait.
 - c. Be aware that most users employ 1024x876 resolution. Don't make diagrams so busy that they are unreadable on the screen, no matter how good they look printed. Break a busy diagram into a leveled set if necessary.
 - d. Factor all these points into your design from the very beginning.
7. Before publishing your template, perform a beta-test of it with a representative sample of your intended audience. Run multiple scenarios if necessary and re-work the template until it is proven user-friendly. This will reduce surprises and frustration for future users.

Word.

8. Use MS Office's document protection features, when appropriate. Some of it is not very intuitive, and beyond the built-in Help, but there are some useful materials on the web.
9. Avoid using macros, which interrupt progress whenever someone opens the document.
10. Avoid using hidden text for instructional data – too many people get confused by this, and you cannot be sure their Word setup even displays hidden text.

11. For instructional text, you may want to create a separate document and embed it into the template. Then it can be double-clicked when needed, or deleted when no longer needed.
12. Don't be amateurish in the use of Word. For example:
 - a. Don't use repeated empty paragraphs to space down in a document. Rather, format the paragraph's 'Before' spacing as needed.
 - b. Don't use repeated tabs or spaces to align data. Use tables or adjust the ruler as needed.
 - c. Learn to use Word's styles appropriately.
13. When appropriate, use Word 2003's advanced features such as:
 - a. Automated Table of Contents generation.
 - b. The control toolbox, which provides checkboxes, radio boxes, drop downs, etc.

[When you use these, however, they add macros, which may not be worth the hassle.]
 - c. Forms and document protection.

PowerPoint.

14. Colors.
 - a. On the master slide, use a white background.
 - b. Use light colors; Use spot color rather than large fills.
15. Test your template by printing a complete sample of it as a 2-up handout. It should still be readable. Many folks, in a spirit of ecology, prefer to print decks this way.
 - a. If you receive a deck from someone that you need to print, go to the master slide first and remove any gratuitous dark backgrounds, logos, or similar unnecessary debris.
 - b. Print on black and white printers whenever you can. Together, let's save toner, especially color -- it's expensive.
16. Maximize screen real estate.
 - a. Get rid of unnecessary filler. We all know we work here -- our logo, for example, doesn't have to be on every slide.
 - b. The confidentiality statement and the document source name could be placed on the first slide only. (The statement we use is this: "Contains U.S. Cellular® confidential information. Not for external use or disclosure without proper authorization.").
 - c. Minimize the graphic clutter. Don't waste the top inch and the left 2 inches with some artistic graphic just because PowerPoint makes that so easy. Use 90-95% of the screen for your data.
 - d. If the template may be used for actual presentations, make sure the default font face and sizes are readable from the back of a typical room.

Excel.

17. Usability.
 - a. Set up worksheet and cell protection as appropriate to simplify use and minimize data entry errors.
 - b. Before the final save, leave the workbook and cursor positioned as you want them to be when opened, since Excel remembers its last state.