Week 12

Chapter 12 - *Automation*

Chapter 13 - *Design Documents*

Chapter 14 - *Oncall*

Last lecture of the semester

Professor Rieks

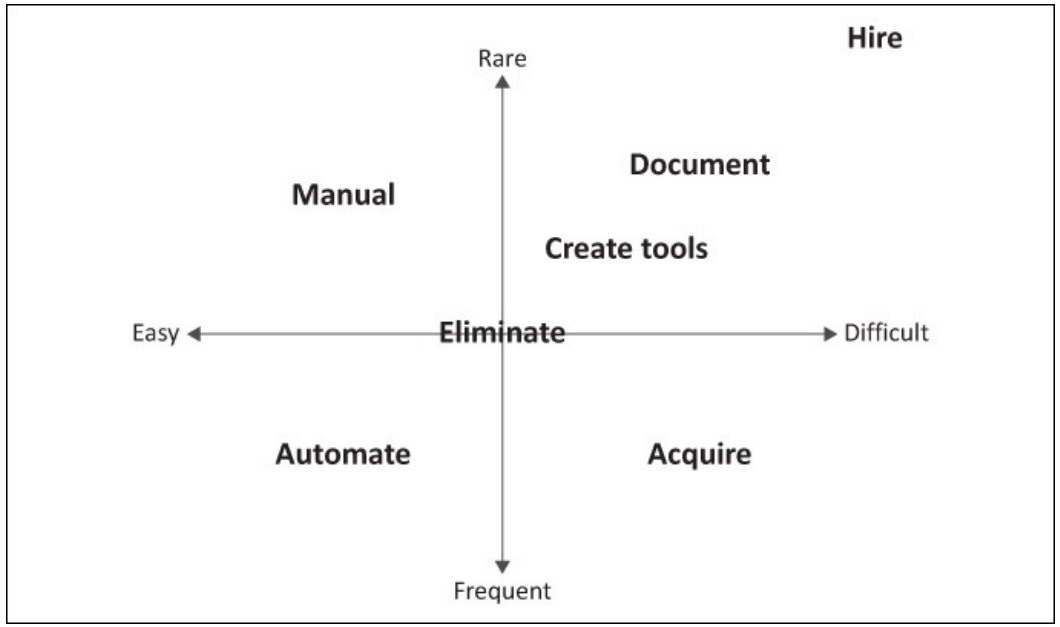
Automation

- Automation is when computers do work for us.
- 3 Approaches to Automation
- “left-over” principle
  - where the automation handles as much as possible, with people being expected to handle whatever is left over.
- “compensatory” principle
  - where the work is divided between people and the automation based on which one is better at which task.
- “complementarity” principle
  - which aims to improve the long-term health of the combined system of people and computers.

“left-over” principle

The x-axis is labeled from “rare” to “frequent,”
The y-axis is labeled from “easy” to “difficult,”
representing how much effort the task requires each time it is performed.

- Tasks classified as rare/easy can remain manual
- Tasks classified as rare/difficult should be documented and tools should be created to assist the process
- Tasks classified as frequent/easy should be automated.
- Tasks classified as frequent/difficult should be automated, but it may be best to acquire that automation rather than write it yourself.



Compensatory Principle

- The compensatory principle is based on Fitts's list, named after Fitts (1951), who proposed a set of attributes to use when deciding what to automate.
- Despite the more than 60 years that have passed since Fitts performed his work, these attributes still apply reasonably well.
- This principle is based on the assumption that the capabilities of people and machines are reasonably static, with the work being divided accordingly.
  - Rather than implicitly considering humans to be infinitely versatile machines, this approach aims to avoid putting excessive demands on people.

Attribute	Machine	Operator/Human
Speed	Much superior.	Comparatively slow, measured in seconds.
Power output	Much superior in level and consistency.	Comparatively weak, about 1500 W peak, less than 150 W during a working day.
Consistency	Ideal for consistent, repetitive actions.	Unreliable, subject to learning (habituation) and fatigue.
Information capacity	Multichannel. Information transmission in megabits/second.	Mainly single channel, low rate < 10 bits/second
Memory	Ideal for literal reproduction, access restricted and formal.	Better for principles and strategies, access versatile and innovative.
Reasoning, computation	Deductive, tedious to program. Fast, accurate. Poor error correction.	Inductive. Easy to program. Slow inaccurate. Good error correction.
Sensing	Specialized, narrow range. Good at quantitative assessment. Poor at pattern recognition.	Wide energy ranges, some multifunction capability. Good at pattern recognition.
Perceiving	Copes poorly with variations in written/spoken material. Susceptible to noise.	Copes well with variation in written/spoken material. Susceptible to noise.

Complementarity Principle

- The complementarity principle looks at automation from the human perspective.
- It aims to help people to perform efficiently in the long term, rather than just looking at short-term effects. It looks at how people's behavior will change as a result of the automation, as well as without the automation.
- In this approach, one would consider what people learn over time by doing the task manually, and how that would change with the automation

Lessons Learned

- The human component may all too often be neglected.
- Systems that aim to eliminate boring and tedious tasks so that people can tackle more difficult tasks leave the hardest parts to humans because they are too complex to be automated. Thus mental fatigue due to many tedious tasks is eliminated, but it is replaced by an even more burdensome mental fatigue due to the need to tackle difficult problems on a continual basis.
- Automation can bring stability to a system, yet this stability results in operators becoming less skilled in maintaining the system. Emergency response becomes particularly brittle.

Hidden Cost of Automation

- Super automated systems often require super training, which can be super expensive.
- Hiring becomes super difficult, which begins to limit the company's ability to grow at its desired rate.
- The missed opportunities that result become a burdensome cost. This opportunity cost may be more expensive than what the system saves.
- Such dilemmas are why companies like Google implement super aggressive recruiting campaigns.

Tool Building versus Automation

- There is a distinction between tool building and automation.
- **Tool building** improves a manual task so that it can be done better.
- **Automation** seeks to eliminate the need for the person to do the task.
- A process is automated when a person does not have to do it anymore, yet this does not eliminate the need for people.
- Once a process is automated, a system administrator's role changes from doing the task to maintaining the automation.
- **Tools Are Good, But Automation Is Better**
- Much of operational work consists of repeated tasks, such as configuring machines, creating accounts, building software packages, testing new releases, deploying new releases, increasing capacity, failing over services, moving services, and moving or reducing capacity.
- All of these tasks can be improved with better tools, and these tools are often stepping stones to automation.
- Automation enables the collection of statistics about failures. If certain situations tend to make the automation fail, those situations can be tracked and investigated.
- Often automation is incomplete and certain edge cases require manual intervention. Those cases can also be tracked and categorized, and the more pervasive ones can be prioritized for automation.

Tool building is good, but automation is required for scalable cloud computing.

Goals of Automation

- **Help scaling.** Automation is a workforce multiplier.
- **Improve accuracy**. Automation is less error prone than people are.
- **Increase repeatability**. Software is more consistent than humans when doing tasks. Consistency is part of a well-controlled environment.
- **Improve reliability**. Once a process is automated, it is easier to collect statistics and metrics about the process.
- **Save time**. There is never enough time to do all the work that needs to be done. An automated task should require less time vs. manual
- **Make processes faster**. Manual processes are slower because they involve thinking and typing. Both are error prone and correcting mistakes often has a large time penalty itself.
- **Enable more safeguards**. Adding additional pre- and post-checks to an automated process is easy. Doing so incurs a one-time cost, but improves the automation for all future iterations of the process.
- **Empower users**. Automation often makes it possible for a non-SA to do a task. Automation turns a task that only an expert can do into one that an end user can do using a self-service tool. .
- **Reduce user wait time.** Manual processes can be done only when an SA is available. Automation can be running all day and all night, and will usually have completed a task before an SA would have been available to start it.
- **Reduce system administrator wait time**. Many manual processes involve doing one step, then waiting some time before the next step can proceed. Computers are better at waiting than people are.

Scripting Languages

- Scripting languages are interpreted languages designed for rapid development, often focusing on systems programming.
- Some common examples include **Perl**, **Python**, and PowerShell.
  - Perl is older and very popular with system administrators because it is similar to C and awk, languages that UNIX system administrators traditionally know.
  - Python has a cleaner design and the code is much more readable than Perl code.
  - PowerShell is new, powerful, and can be extended to various situations and products through a vendor provided API
- Compiled languages can be a good choice for large-scale automation. Automation written in a compiled language typically scales better than the same automation written in a scripting language.
- Compiled languages often used by system administrators include C, C++,

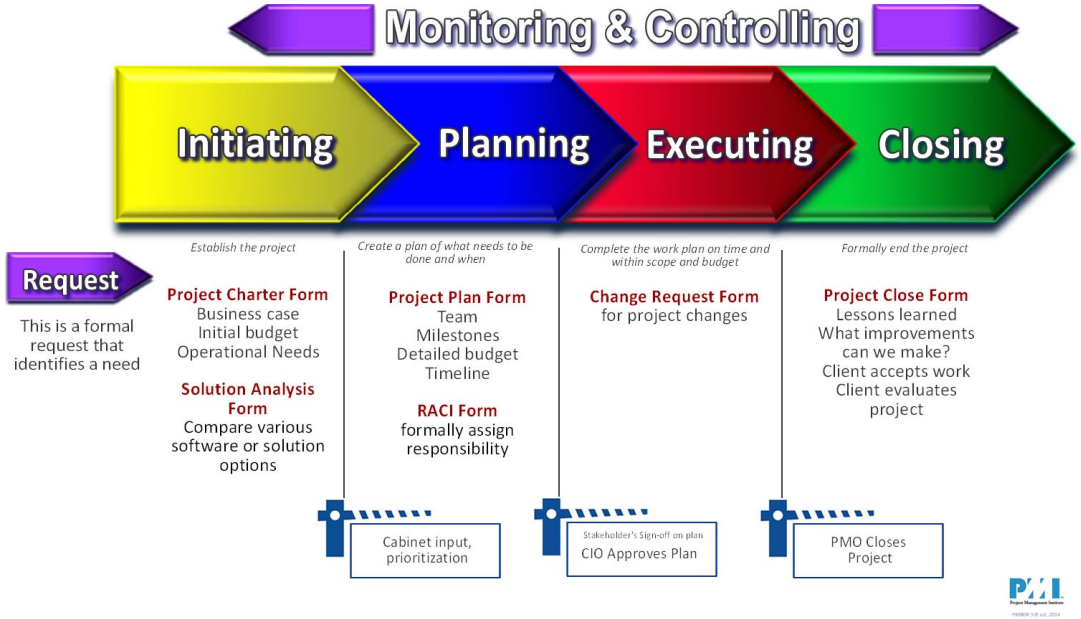
Design Documents

- Design documents are written descriptions of proposed and completed projects, big or small. They serve as a roadmap for your projects and documentation of your accomplishments. They are descriptions of proposed or completed projects.
- They record the goals, the design itself, alternatives considered, plus other details such as cost, timeline, and compliance with corporate policies.
- Writing out what you are going to do forces you to think through the details.
- Having a written document can make design collaborative when the document becomes a communication vehicle for ideas.
- Written documents mean fewer surprises for your teammates, and they help you to get consensus among the team before moving forward. After the project is completed, the design document serves as an artifact that documents the work, a reference for the team.
- Design documents can also be used to describe changes instead of projects.

Project Management Institute

- **Initiating** In the initiation phase, you'll define the project. You'll sort out the project goals, scope, and resources of the project, and what roles are needed on the team. Clarifying what stakeholders expect out of the project, and what exactly the project is aiming to achieve (and why) will give the project and team clear direction
- **Planning**. In the planning phase, you'll determine the steps to actually achieving the project goals—the "how" of completing a project. You'll establish budgets, timelines, and milestones, and source materials and necessary documents. This step also involves calculating and predicting risk, putting change processes into place, and outlining communication protocols. If the initiation phase is assembling your troops, the planning phase is deciding what to do with them.
- **Execute**. Executing a project means putting your plan into action and keeping the team on track. Generally, this means tracking and measuring progress, managing quality, mitigating risk, managing the budget, and using data to inform your decisions.
- **Close projects**. In the closing phase of the project management lifecycle, you'll conclude project activities, turn the finished product or service over to its new owners, and assess the things that went well and didn't go so well. It'll also be a time to celebrate your hard work.

PMP Forms.zip



https://www.coursera.org/articles/project-management-lifecycle

Designing Oncall

- Oncall is the practice of having a group of people take turns being responsible for exceptional situations, more commonly known as emergencies or, less dauntingly, alerts.
- Oncall schedules typically provide 24 × 7 coverage. By taking turns, people get a break from such heightened responsibilities, can lead normal lives, and take vacations.
- When an alert is received, the person on call responds and resolves the issue, using whatever means necessary to prevent SLA violations, including shortcut solutions that will not solve the problem in the long term.
  - If he or she cannot resolve the issue, there is an escalation system whereby other people become involved. After the issue is managed, any follow-up work should be done during normal business hours—in particular, root causes analysis, postmortems, and working on long-term solutions.

Service Level Agreement (SLA) and Roster

- When designing an oncall scheme for an organization, begin with the SLA for the service. Work backward to create an SLA for oncall that will result in meeting the SLA for the service. Then design the oncall scheme that will meet the oncall SLA.
  - For example, suppose a service has an SLA that permits 2 hours of downtime before penalties accrue. Suppose also that typical problems can be solved in 30 minutes, and extreme problems take 30 minutes to cause system failover but usually only after 30 minutes of trying other solutions. This would mean that the time between when an outage starts and when the issue is being actively worked on must be less than an hour.
- **Oncall Roster**: The roster is the list of people who take turns being oncall.
  - The list is made up of qualified operations staff, developers, and managers. All operations staff should be on the roster. This is generally considered part of any operations staff member's responsibility.

Oncall Schedule Design

Weekly: A person is oncall for one week at a time. The next shift starts the same time each week, such as every Wednesday at noon. Having the change occur mid-week is better than during the weekend. If the change happens on a Wednesday, each onduty person has one complete weekend where travel and other fun are limited.

- **Daily**: A person is oncall for one day at a time. This may seem better than a weekly schedule because the shift is not as long, but it means being oncall much more often. A weekly schedule might mean being oncall one week out of every six. With a small team, a daily schedule might mean being oncall every six days, never having a complete week to take a vacation.
- **Split Days**: On a given day multiple people are oncall, each one responsible for a different part of the day or shift. For example, a two-shift schedule might involve two 12-hour shifts per day. One person works 9 AM to 9 PM and another is oncall for the overnight. This way, if an alert happens in each shift, someone is always able to sleep. A three-shift schedule might be 8 hours each: 9 AM to 5 PM, 5 PM to 1 AM, and 1 AM to 9 AM.
- **Follow the Sun**: Members of the operations team live in different time zones, and each is oncall for the hours that he or she would normally be awake (sun-light hours).
    - If the team resides in California and Dublin, a shift change at 10AM and 10PM California time means all members have some responsibilities during office hours and sleeping hours, plus there are enough overlap hours for inter-team communication.

Compensation

- Oncall Compensation drives some design elements of the schedule. In some countries, being oncall requires compensation if response time is less than a certain interval.
- The compensation is usually a third of the normal hourly salary for any hour oncall outside of normal business hours (can vary greatly). It may be paid in cash or by giving the oncall person time off.
- Compensation rates may be different if the person is called to action.
- In some countries, there is no legal obligation for oncall compensation but good companies do it anyway because it is unethical otherwise.
- One benefit of follow-the-sun coverage is that it can be constructed in a way that maximizes time oncall during normal business hours for a location, while minimizing the amount of additional compensation that needs to be budgeted.

OnCall Notifications

- There are many levels of urgency at which monitoring and other services need to raise the attention of human operators. Only the most urgent is an alert.
- Each level of urgency should have its own communication method. If urgent alerts are simply sent to someone's email inbox, they may not be noticed in time. If non-urgent messages are communicated by sending an SMS to the person oncall, the "*Boy Who Cried Wolf*" syndrome will develop.
- The best option is to build a very high-level classification system:
  - **Alert Oncall**: The SLA is in violation, or if a condition is detected that, if left unattended, will result in an SLA violation.
  - **Create a Ticket**: The issue needs attention within one business day.
  - **Log to a File**: The condition does not require human attention. We do not want to lose the information, but we do not need to be notified.
  - **Do Nothing**: There is no useful information; nothing should be sent.

  Email is, quite possibly, the worst alerting mechanism. Expecting someone to sit and watch an email inbox is silly, and a waste of everyone's time. With this strategy, staff will be unaware of new alerts if they step away or get involved in other projects.

Alerts

- Once an alert is triggered, there are many ways to notify the person who is oncall.
  - **One-Way and Two-Way Pagers**: Hand-held devices that receive text messages. Two-way pagers permit sending a reply to acknowledge that the message was received.
  - **SMS or Text Message to a Mobile Phone**: Sending a text or SMS message to a person's mobile phone is convenient. In some countries, pagers are significantly more reliable than SMS. If you are creating an alerting system for co-workers in another country, do not assume that what works well for you will be viable elsewhere. Local people should test both.
  - **Smart Phone App**: Smart phone apps are able to display additional information beyond a short text message. However, they often depend on Internet connectivity, which may not always be available.
  - **Voice Call**: A voice synthesizer and other software is used to call a person's phone and talk to him or her, asking the person to press a button to acknowledge the message (otherwise, the escalation list will be activated).
  - **Chat Room** Bot: A chat room bot is a software robot that sits in the team's chat room and announces any alerts. This is a useful way to keep the entire team engaged and ready to help the oncall person if needed.
  - **Alerting Dashboard**: The alerting dashboard is a web page that shows the history of alerts sent. It provides useful context information.
  - **Email**: Email should never be the only way the oncall person is alerted.
    - Sitting at your computer watching your inbox is a terrible use of your time. Nevertheless, it is useful to have every alert emailed to the oncall person as a backup method. This way the full message is received; SMS truncates messages to 160 characters.

Oncall Playbook

- Ideally, every alert that the system can generate will be matched by documentation that describes what to do in response. An oncall playbook is this documentation.
- The general format is a checklist of things to check or do. If the end of the list is reached, the issue is escalated to the oncall escalation point (which itself may be a rotation of people).
- This creates a self-correcting feedback loop. If people feel that there are too many escalations waking up them late at night, they can correct the problem by improving the documentation to make oncall more self-sufficient.
- If they feel that writing documentation is unimportant or “someone else’s job,” they can, by virtue of not creating proper checklists, give oncall permission to wake them up at all hours of the night.
- It is impressive how someone who feels that writing documentation is below them suddenly learns the joy of writing after being woken up in the middle of the night. The result of this feedback loop is that each checklist becomes as detailed as needed to achieve the right balance.

Periodic Review of Alerts

- The alert log should be reviewed periodically to spot trends and allocate resources to create long-term fixes that ultimately reduce the total number of alerts received.
- There should be a systematic approach to reduce the number of alerts or entropy is likely to make
- It’s useful to have a weekly meeting to review alerts and issues and look for trends.
- The alert log should be annotated by the person who received the alert. Most systems permit alerts to be tagged with keywords. The keywords can then be analyzed for trends.

Questions