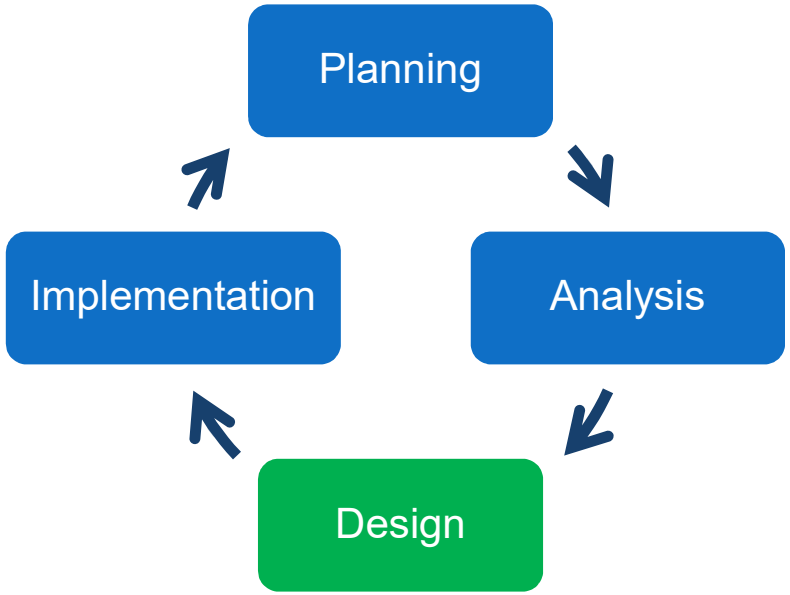
MOVING TO DESIGN – DESIGNING THINKING AND DESIGNING AGAINST FAILURE

Dr. Joycelyn Streator

Topics

- Design Thinking
- Designing IS – General Considerations
- Mistake Proofing in IS
- Failure Mode Effect Analysis

- Once we have gathered requirements and analyzed the as-is system, the systems project moves to the Design phase.
- What is Design?
- How do you Design?



Design Thinking

- Design Thinking, a solution-based approach to finding what would-be users *really* need.

- Design thinking's tools and methods borrow from a variety of disciplines, including ethnography, computer science, psychology and organizational learning.

- Design teams use Design Thinking to tackle ill-defined or unknown problems (wicked problems) because it reframes these types of problems in *human-centric* ways, allowing the designer to focus on what's most important for users/customers.

Design Thinking

- Stanford's Hasso-Plattner Institute of Design describes Design Thinking as a five-stage process.

- These stages are not always sequential; nor do they require design teams to approach them thus. Designers will find the stages often occur in parallel and see repeated use on an iterative basis.

- Consequently, designers should consider these not a "recipe", but different *modes* that contribute to a project.

Empathize

- The first stage of the Design Thinking process demands gaining an empathic understanding of the problem you're trying to solve, typically through some form of user research.

- Empathy is crucial because it allows you to set aside your own assumptions about the world in order to gain insight into users and their needs. This stage involves entering the realm of the users and, as far as possible, "becoming" them so as to begin work on custom-designing a solution.

Define

- During the Define stage of Design Thinking, you put together the information you have created and gathered during the Empathize stage.

- You analyze your observations and synthesize them in order to define the core problems you and your team have identified so far. This is where you ensure that what you are addressing sits in sharp relief before you, its properties known in full.

Ideate

- The process's third stage finds you ready to start generating ideas. With the knowledge you have gathered in the first two phases, you can start to **"think outside the box"** to identify new solutions to the problem statement you've created, and you can start to look for alternative ways of viewing the problem.

Prototype

- In the Prototype phase of Design Thinking, your design team produce a number of inexpensive, scaled-down versions of the product or specific features found within the product so you can investigate the problem solutions generated in the previous stage.

Test

- In the Test phase of Design Thinking, you rigorously test the completed product using the best solutions identified during the prototyping phase.

- This is the final stage; however, in an iterative process, the results generated during the testing phase are what you will often use to redefine one or more problems.

Design Thinking – Ted Talk

- https://www.youtube.com/watch?v=UAinLaT42xY

DESIGNING AGAINST FAILURE

Designing Against Failure

Thorough design of Information Systems should consider

- What key information or features are needed (requirements)
- What can be eliminated (process efficiency)
- How to build in safeguards against failure

Designing IS– General Considerations

What key pieces of information do users of business process software need?

Relevant Data

Process Progress

Urgency & Timeliness

List of Options Available of Actions

Data forms with calculations “baked-in” whenever feasible

Data Validation

Conditional Input Fields

Concept of Mistake Proofing

- A strategy for preventing errors in processes
- Makes it impossible for defects to pass unnoticed and corrects them immediately
- Technique detects defects and prevents them from moving into next area

Everyday Examples of Mistake Proofing

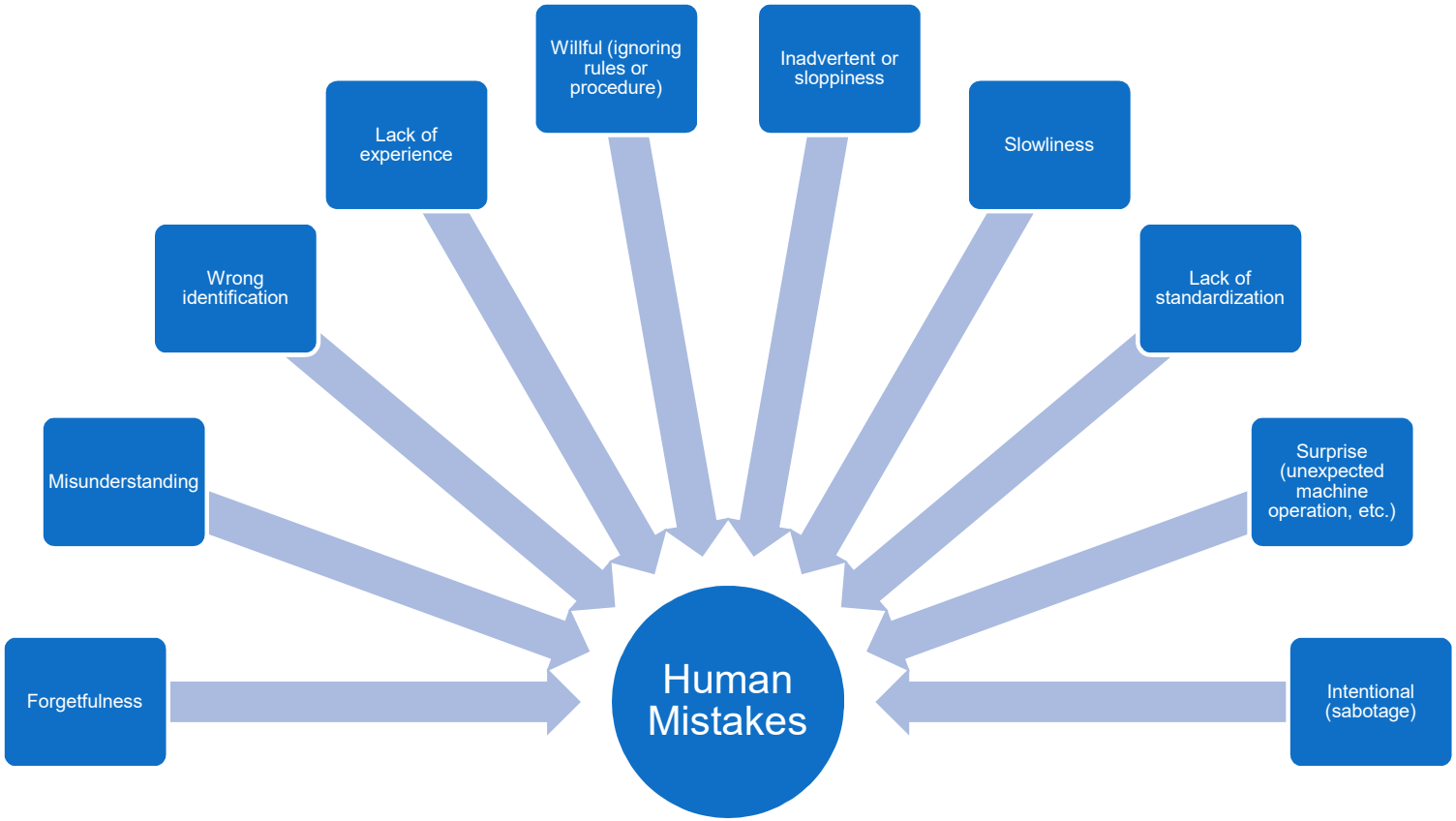
- Home
  - ✓ auto shut off on iron

- Automobile
  - ✓ car won't start unless transmission is in park/neutral

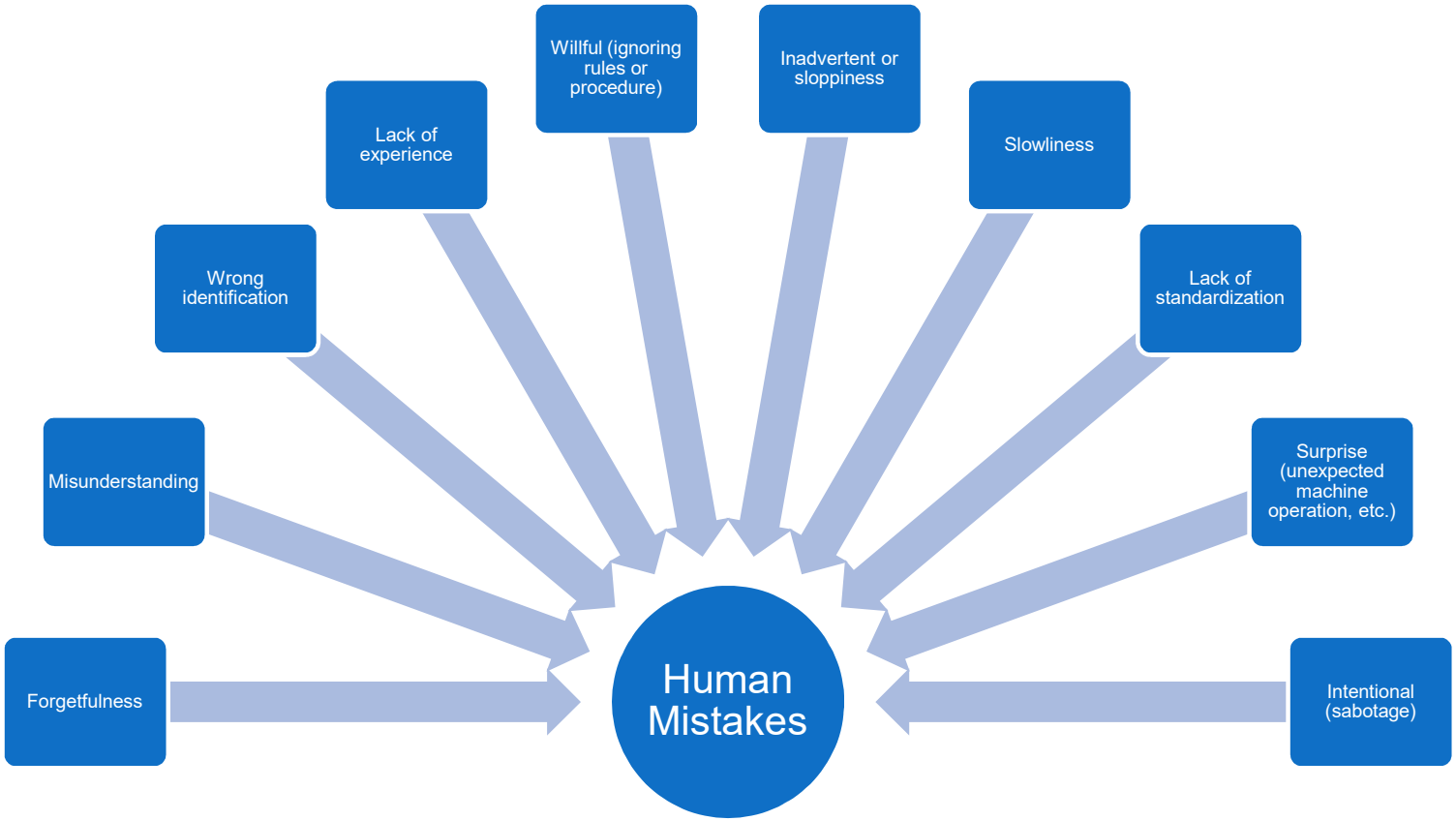
Benefits of Mistake Proofing

- Improved quality and customer satisfaction
- Prevents error occurrence
- Prevents defect occurrence
- If defects occur, ensures effective action
- Little or no formal training
- Cost effective
- Easy to implement

Ten Types of Human Mistakes



Ten Types of Human Mistakes



General IT Mistake Proofing Guidelines

Guidance	• When someone runs into a rule preventing an action like an access control limit, give clear guidance through the user interface on how to get permission, the proper process and/or how to contact an expert for guidance.
Simplicity	• Design processes to be as simple as possible. Simplify existing flows as much as possible.
Flexibility	• Wherever possible, give people the ability to go back in the process if they accidentally start the wrong one.
Fit	• Review your process flows to make sure they closely fit business processes instead of becoming a hindrance to them.

General IT Mistake Proofing Guidelines

Data	• Demand as little data as possible from users, but make essential fields mandatory to progress.
Intentional	• Design controls and interfaces to minimize accidental selections of the wrong thing.
Uncluttered	• Have a clean, uncluttered user interface and reports so people can quickly find what they want.
Indicators	• Have clear indicators when something is wrong. Think of the example of statistical process control charts or green/red indicators on system health dashboards.

Mistake Proofing Interfaces - 1

Prevent actions and disable elements	Forms are often seen as a default practice, but they are ideal for applying Poka-Yoke. Preventing an action to appear until all of the conditions necessary to perform that action are met is a great example. For instance, don't allow to proceed with a registration form until all required fields are filled in. Disable submit button if input field is empty. Disable Delete all link if less than two items in the list are selected. Examples are endless
Use defaults	If a user wants to order an accessories from an fashion e-commerce, you can set 1 as the default value (instead of leaving the field empty that would trigger an error). Default settings are a great asset in error prevention

Mistake Proofing Interfaces - 2

Using constraints	You can restrict user's inputs allowing for instance only numeric characters, or disallowing special characters. This enhance the experience but error proofing the inputs. To improve even more the experience, the application my show an usage example to explain how to fill in the form or complete a designated task.
Restrict user's input	You can restrict user's input in many ways – allow only numeric characters, disallow special characters, limit user's input to a specific range or limit the size of user's input. This is usually done by rejecting user's input if invalid characters are detected. You should always make clear to the user why rejection happened (e.g. "You can enter only numbers").

Mistake Proofing Interfaces - 3

Mask user's input	Similar to previous technique, masking forces users to enter information in a specific format. It is a good practice to reveal mask in input field upfront and make it clear to the user which format is allowed. The problem with masking is that help text is not always clear enough – would users know what "~9.99 ~9,999.99" means? In case when masking pattern is not easily understandable, it is better to show users an example of usage.
Correct user's input	In some cases, the application will be able to correct the user's input automatically without interrupting her/him. For instance, the application can automatically add "http://" to an URL if user omitted it. Another example is when user enters a value that is larger than maximum allowed. In this case the application can replace the user's input with maximum allowed value while letting the user accept it or change to another value.

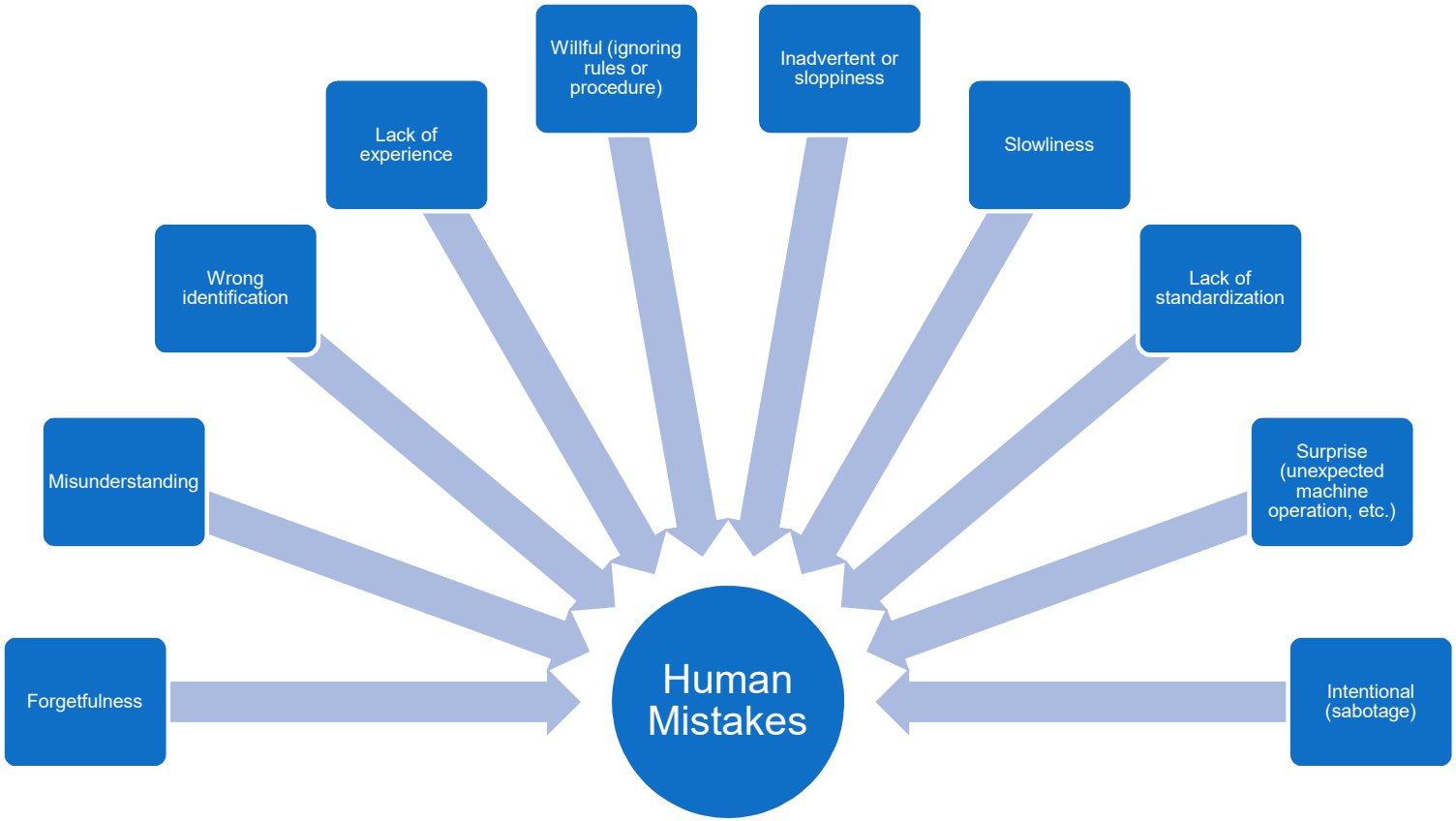
Mistake Proofing Interfaces - 4

Use defaults	In some cases defaults can help further in error prevention. If user wants to book a flight and have to choose a number of passengers, you can set 1 as default (instead of leaving this field empty which will trigger validation later). Another good example I can think of is my android-based mobile phone. Each time user starts a new SMS message, upper-case button is turned on. When user types the first letter it turns off.
Use appropriate controls	You can restrict user's inputs allowing for instance only numeric characters, or disallowing special characters. This enhance the experience but error proofing the inputs. To improve even more the experience, the application my show an usage example to explain how to fill in the form or complete a designated task.

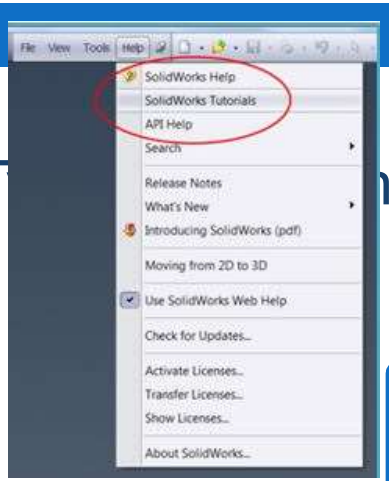
Mistake Proofing Interfaces - 5

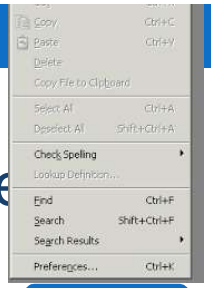
Miller's Law	In theory, especially in context of computers, humans too have RAM. According to psychologist George Miller, the maximum number of objects that a human can hold in his working memory is 7. In the context of UID, this translates to the notion that no interface should have more than 7 active elements at any time. Less is of course better, but more than that will tend to confuse a user. Again, the concept emphasized here is simplicity over complexity.
Signal-to-Noise Ratio	With UID, "Signal" elements refer to content of basic importance (the information that will be of most crucial value to users), while "Noise" elements refer to everything else (superfluous information). A good UID focuses on emphasizing the Signal rather than the Noise.

Ten Types of Human Mistakes



Ten T… n Mistake…





Lack of experience

Willful (ignoring rules or procedure)

Inadvertent or sloppiness

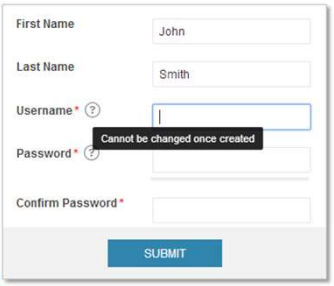
Slowliness

Wrong identification

Lack of standardization

Misunderstanding

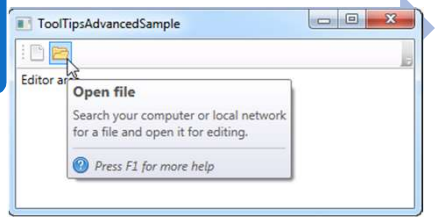
Surprise (unexpected machine operation, etc.)



Forgetfulness

Human Mistakes

Intentional (sabotage)



FAILURE MODE EFFECT ANALYSIS

FMEA

- A structured approach to:
  - Identifying the ways in which a product or process can fail
  - Estimating risk associated with specific causes
  - Prioritizing the actions that should be taken to reduce risk
  - Evaluating design validation plan (design FMEA) or current control plan (process FMEA)

What Is A Failure Mode?

- A **Failure Mode** is:
  - The way in which the component, subassembly, product, input, or process could fail to perform its intended function
    - Failure modes may be the result of upstream operations or may cause downstream operations to fail
  - Things that could go wrong

FMEA

- **Why**
  - Methodology that facilitates process improvement
  - Identifies and eliminates concerns early in the development of a process or design
  - Improve internal and external customer satisfaction
  - Focuses on prevention
  - FMEA may be a customer requirement (likely contractual)
  - FMEA may be required by an applicable Quality Management System Standard (possibly ISO)

Application Examples

- Manufacturing: A manager is responsible for moving a manufacturing operation to a new facility. He/she wants to be sure the move goes as smoothly as possible and that there are no surprises.
- Design: A design engineer wants to think of all the possible ways a product being designed could fail so that robustness can be built into the product.
- Software: A software engineer wants to think of possible problems a software product could fail when scaled up to large databases. This is a core issue for the Internet.

When to Conduct an FMEA

- Early in the process improvement investigation
- When new systems, products, and processes are being designed
- When existing designs or processes are being changed
- When carry-over designs are used in new applications
- After system, product, or process functions are defined, but before specific hardware is selected or released to manufacturing

The FMEA Form

Process/Product
Failure Modes and Effects Analysis Form
(FMEA)

Processor Product Name:		Prepared by:	Page ___ of ___
Responsible:		FMEA Date (Orig) _____ (Rev) _____	

Process Step / Input	Potential Failure Mode	Potential Failure Effects	SEVERITY	Potential Causes	OCCURRENCE	Current Controls	DETECTION	RPN	Actions Recommended	Resp.	Actions Taken	SEVERITY	OCCURRENCE	DETECTION	RPN
What is the process step and Input under investiga-tion?	In what ways does the Key Input go wrong?	What is the impact on the Key Output Variables (Customer Requirements)?		What causes the Key Input to go wrong?		What are the existing controls and procedures (inspection and test) that prevent either the cause or the Failure Mode?			What are the actions for reducing the occurrence of the cause, or improving detection?		What are the completed actions taken with the recalculated RPN?				
								0							0
								0							0
								0							0
								0							0
								0							0

Identify failure modes and their effects

Identify causes of the failure modes and controls

Prioritize

Determine and assess actions

FMEA Procedure

1. For each process input (start with high value inputs), determine the ways in which the input can go wrong (failure mode)
2. For each failure mode, determine effects
   - Select a severity level for each effect
3. Identify potential causes of each failure mode
   - Select an occurrence level for each cause
4. List current controls for each cause
   - Select a detection level for each cause

FMEA Procedure (Cont.)

5. Calculate the Risk Priority Number (RPN)
6. Develop recommended actions, assign responsible persons, and take actions
   - Give priority to high RPNs
   - MUST look at severities rated a 10
7. Assign the predicted severity, occurrence, and detection levels and compare RPNs

Severity, Occurrence, and Detection

- Severity
  - Importance of the effect on customer requirements
- Occurrence
  - Frequency with which a given cause occurs and creates failure modes (obtain from past data if possible)
- Detection
  - The ability of the current control scheme to detect (then prevent) a given cause (may be difficult to estimate early in process operations).

Rating Scales

- There are a wide variety of scoring “anchors”, both quantitative or qualitative
- Two types of scales are 1-5 or 1-10
- The 1-5 scale makes it easier for the teams to decide on scores
- The 1-10 scale may allow for better precision in estimates and a wide variation in scores (most common)

Rating Scales

- Severity
  - 1 = Not Severe, 10 = Very Severe
- Occurrence
  - 1 = Not Likely, 10 = Very Likely
- Detection
  - 1 = Easy to Detect, 10 = Not easy to Detect

Risk Priority Number (RPN)

- RPN is the product of the severity, occurrence, and detection scores.

EXERCISE: DESIGNING AGAINST FAILURE

EXERCISE

- A motorist was traveling alone in a remote community. The vehicle veered off the road got stuck in a ditch on a back road. The motorist contacted AAA motor club to tow the vehicle, become disoriented and was unable to explain his exact location to the company dispatch.

- After this incident. AAA developed a free phone iPhone App to address this type of situation. Using this mobile application, a stranded motorist can quickly request roadside assistance at the tap of an icon. A message that includes the driver's membership ID, along with their iPhone location data, is automatically composed and forwarded to the AAA dispatch office. Once the dispatcher receives the message, assistance is sent to the coordinates. The app eliminates the need to call a number, wait on hold, recite a membership number, or explain a geographical location. However, it does allow the driver the option of speaking with the dispatcher, request 911 emergency assistance, or identify the closest AAA-approved auto repair locations.

- You are tasked with refining the design by identifying potential failure modes.

Process Model