Queen Mary
University of London

EBU5608 Product Development and Management

Topic 12 – Production ramp-up, Robust Design

Queen Mary
University of London

Agenda

- Production ramp-up
- Robust design
- Robust design process
- Software robustness

Queen Mary
University of London

Phase 5 – Production ramp-up

Product Development Process

- The 6 distinct phases of product development are:
  - 0. **Planning**
  - 1. **Concept development**
  - 2. **System-level design**
  - 3. **Detail design**
  - 4. **Testing and refinement**
  - 5. **Production ramp-up**

4

4

Phase 5 – Production ramp-up [1]

- In the production ramp-up phase, the product is made using the **intended production** system
- The purpose of the ramp-up is to **train** the work force and to work out any **remaining problems** in the production **processes**
- **Products** produced during production ramp-up are sometimes supplied to **preferred customers** and are carefully **evaluated** to identify any remaining **flaws**

Phase 5 – Production ramp-up (cont.)

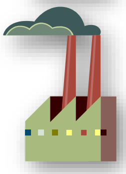
- The **transition** from production ramp-up to on-going production is usually **gradual**
- At some point in the transition, the product is **launched** and becomes available for widespread distribution
- This is the final stage of the development process, however remember that **iteration** can take place at any point
- Some activities also take place in a number of phases
  - robust design is a example of this

6

6

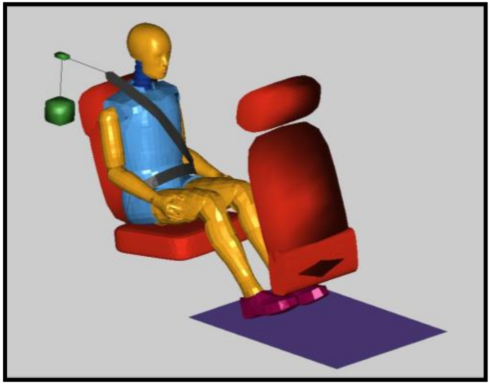
Queen Mary
University of London

Departmental responsibilities



- Departmental responsibilities at this phase
  - **Marketing**
    - Place early production with key customers
  - **Design**
    - Evaluate early production output
  - **Manufacturing**
    - Begin operation of entire production system

Robust Design



8

8

Queen Mary
University of London

What is 'Robust Design'

- A **robust product** is one that has **robust performance**, i.e. it performs **as intended** even under **non-ideal** conditions
  - e.g. manufacturing variations, varying operating conditions
- **Robust design** is the product development activity of creating a robust product
- We need to carry out **tests** or experiments at various stages in the design process to help us create this robust product

Queen Mary
University of London

What is robust design (contd.)

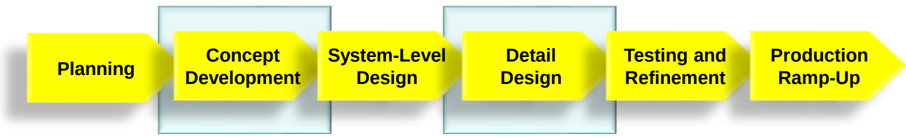
- For a given performance target, many **combinations** of parameter values can give us the desired result
  - e.g. you can compensate for a mobile phone case attenuating the radio signal by increasing the gain of the RF amplifier
- Some of these parameters are **more sensitive** to uncontrolled variables than others
- We need to choose the set of parameters that gives us the **lowest overall sensitivity** and meets all of our **design criteria**
  - This is known as a **robust setpoint**

10

10

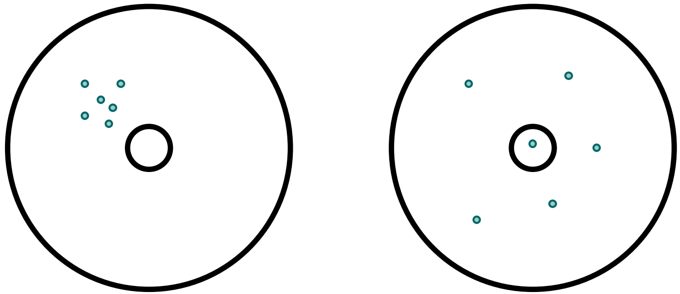
Where does this happen in product development?

- The **earlier** that robustness can be considered in product development, the **better the results**
- Robust design experiments can be used in **concept development** to refine the specifications
- Used most frequently during the **detail design phase** to make sure that we get the required product performance under a variety of conditions



Queen Mary
University of London

Who is the better target shooter?

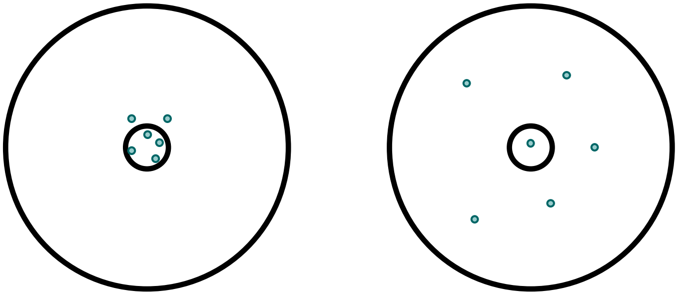


Sam John

12

12

Who is the better target shooter?



Sam

Sam can simply adjust his sights.

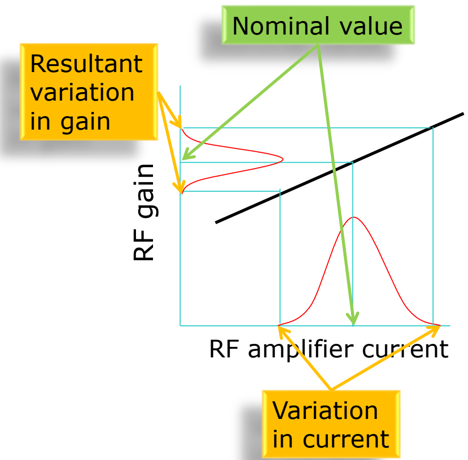
John

John requires lengthy training.

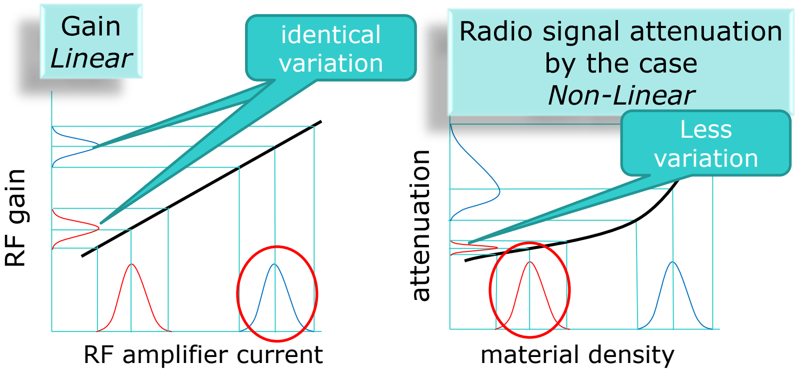
Parameter values and variation

An example:

- A specific amplifier current may give us a specific gain
- However, the value of that current may vary from its nominal value due to, e.g., power supply manufacturing variations
- This will result in us seeing some variability in the RF gain



The sensitivity of setpoints



- The strength of the RF signal is, for example, a function of the gain and the attenuation
- We would choose setpoints based on a combination of values that gave the right strength and least variation

Queen Mary
University of London

The robust design process

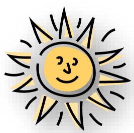
Identify control factors, noise factors & performance metrics
↓
Formulate an objective function
↓
Develop the experimental plan
↓
Run the experiment
↓
Conduct the analysis
↓
Select & confirm factor setpoints
↓
Reflect & repeat

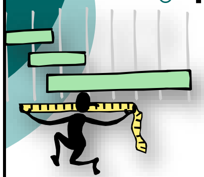
16

16

1. Identify control factors, noise factors & performance metrics

- **Control factors**
  - The design variables to be varied in a **controlled** manner during the **experiment** – e.g. amplifier current
  - Normally these are factors that can be **specified** for production and/or operation
- **Noise factors**
  - The variables that often **cannot** be **controlled** in production and/or operation – e.g. ambient temperature
  - These changes, which may not be controllable, and their effect are monitored during the experiment





1. Identify control factors, noise factors & performance metrics

- **Performance metrics**
  - The product specifications that we are interested in during the experiment – e.g. overall RF signal strength
  - These will be taken from the product specification created in Phase 1 – Concept Development
- After listing the **possible factors**, the team then has to decide which ones will be tested in the experiment
  - This may be done by running **initial models** to see which factors are likely to have the **greatest impact** on the performance metrics

18

2. Formulate the objective function

- The performance metrics are then stated as **objective functions**
- These may be functions giving values that you want to:
  - **Maximise** – e.g. energy efficiency
  - **Minimise** – e.g. waste heat
  - Meet a **target** value – e.g. signal strength
  - Meet a **signal-to-noise ratio** - i.e. reduce the response to uncontrollable factors compared to the response to controllable factors

Queen Mary University of London

3. Develop the experimental plan

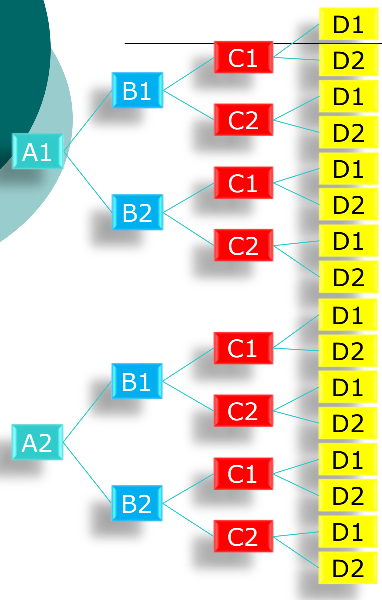
- A critical factor in any experiments is the **cost**
- Where the cost is **low** it may be possible to
  - Run a large number of trials
  - Explore many factor combinations and interactions
- Where the cost is **high**, methods must be used that **reduce** the number of trials by simultaneously changing several factors
- These two cases are shown in the next two slides

20

20

Queen Mary University of London

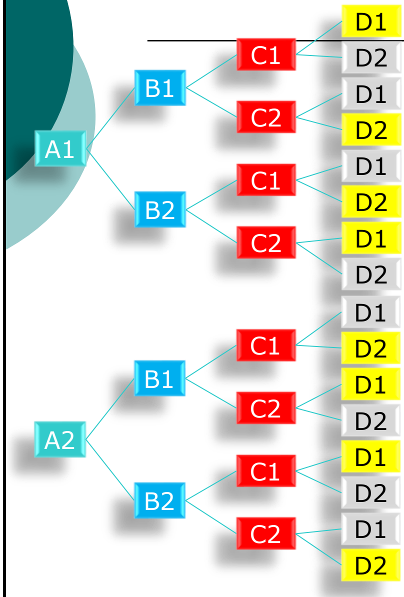
Experimental plan – the number of trials



- A **full factorial** set of trials involves testing a number of levels (e.g. 1, 2) of each factor (e.g. A, B, C, D)
- The number of trials will be L^F , where L is the number of levels of each of the factors F
- In this case we have 2 levels of each of the factors A,B,C,D
  - i.e. 16 trials
- This can be very high
  - e.g. 7 factors at 2 levels would need 128 trials

21

Experimental plan – reducing the number of trials



- A **fractional factorial** set of trials involves testing a statistically selected **subset** of the full trials
- This subset needs careful selection so that it tests **each** of the factors to the **same degree**
- The diagram shows a **½ fractional** set. Much smaller sets are possible, e.g. $^{1}/_{16}$ fractional

Queen Mary University of London

4. Run the experiment

- The product is tested under the conditions described by each combination of factors in the experimental plan
- Wherever possible, the sequence of events is **randomised** to overcome the effect of any factors drifting over time; e.g. **without** randomisation:
  - The first half of the tests use factor A at value 1 and the second half use factor A at value 2
  - There is a drift over time in some other factor such as room temperature
  - Any change in the test results caused by this temperature change would be wrongly attributed to a change from A1 to A2

5. Conduct the analysis

- Statistical techniques are used to analyse the test results to determine which **set of values** for each of the **control** factors will give us the result that is **closest** to the **objective functions** defined in Step 2
- It is important to take into account the **range** of results found for each factor, since it is this variance that is likely to be found, and cause problems, in practice

24

Queen Mary University of London

6. Select & confirm factor setpoints

- The analysis in Step 5 helps the team to understand which factors are best able to **reduce** the product's **variance**
- By choosing setpoints based on this analysis, the product can be designed to give greater robustness

7. Reflect and repeat

- One round of tests may be enough to achieve the objectives
- Sometimes, further refinement is needed with more testing. If this happens, the team needs to:
  - Reconsider the **setpoints** where there is a trade-off of performance versus robustness
  - Explore the **interactions** between some of the factors
  - Consider revising the **ranges** of values tested

26

26

Queen Mary
University of London

Software robustness

- ***Software robustness*** is an important characteristic of any high-quality software system, regardless of its purpose or complexity.
- Without robustness, software is prone to crashes, errors, and vulnerabilities, which can compromise the functionality, security, and reliability of the system.
- It refers to the ability of a software system to continue functioning correctly and reliably even in the face of unexpected or abnormal inputs or situations. These might include things like incorrect user input, network failures, hardware malfunctions, or even deliberate attempts to disrupt the system.

Queen Mary
University of London

Robust Software Application

A robust software application is one that is able to

- Handle a wide range of inputs, including those that are unexpected or incorrect.
- Recover from errors or failures gracefully, without causing data loss or system instability.
- Manage large amounts of data and complex operations without slowing down or crashing.
- Maintain the integrity and consistency of data, even in the face of unexpected events, failures or external stresses, over a long period of time.
- Protect against security threats and vulnerabilities, such as hacking, malware, or unauthorized access.

Queen Mary
University of London

Benefits of Robust Software

The benefits of building robust software include:

- Increased **reliability** and **consistency** of the application
- Reduced **downtime** and maintenance **costs**
- Improved **user experience** and **satisfaction**
- Enhanced **security** and **protection against cyber threats**
- Increased **trust** and **confidence** in the software

29

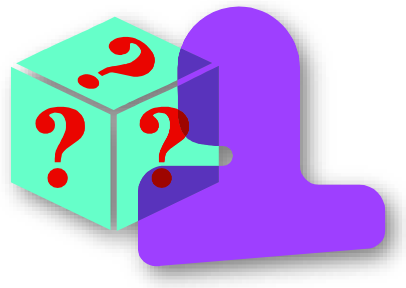
Summary

Robust design is a set of engineering design methods used to create robust products and processes.

Software robustness is a critical quality attribute for any software system. By ensuring that software can continue to operate correctly and reliably even in the face of unexpected situations or inputs, robustness helps to ensure the safety and reliability of the system.

Queen Mary
University of London

Questions?



Go to www.menti.com to post your questions

Mentimeter

Queen Mary
University of London

References

- Product Design and Development, Karl T Ulrich and Steven D Eppinger, 7th Edition (7^{th}) McGraw-Hill, Chapter 15. Robust Design, pages 317 – 331

- What is Software Robustness? How to Ensure Quality and Reliability, Nexus Software Systems, 1995-2023, https://nexwebsites.com/blog/software-robustness/

Queen Mary University of London

Reading

- **Core Textbook** (Ulrich & Eppinger, 7th Edition)
  - Chapter 15. Robust Design

    pages 317 - 331

34