What are use cases and why are they helpful?

INCOSE IW MBSE Workshop 2015

Chris Unger
GE Healthcare
Christopher.unger@med.ge.com

imagination at work

Bottom Line Up Front

Requirements interact with each other

- Some interactions are driven by system interfaces and other aspects of the system design
- Other interactions are driven by the clinical environment
- Still others emerge as a result of how the clinical environment impacts the behavior of the system
- All of these interactions affect the user and the patient experience

The systems engineering process provides methods to examine how the system and the clinical environment drive behavior and user experience

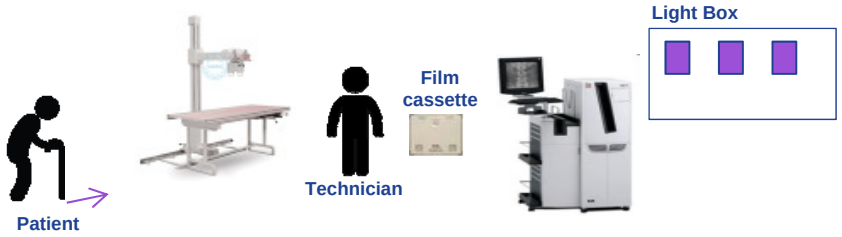
- Context diagrams can capture interactions
- Use cases can capture user and patient experience
- Combining these with conventional requirements analyses yields a better product

The following example describes a medical device, but the methodology applies more broadly to all healthcare processes

A Parable of Requirements Management

Challenge: New “Portable” version of a product

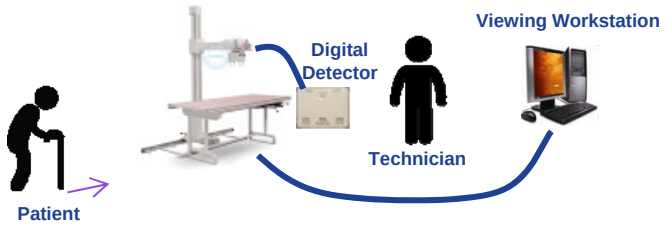
Film



+ Simple and cheap, portable

- Poor IQ, poor workflow (no preview), chemicals

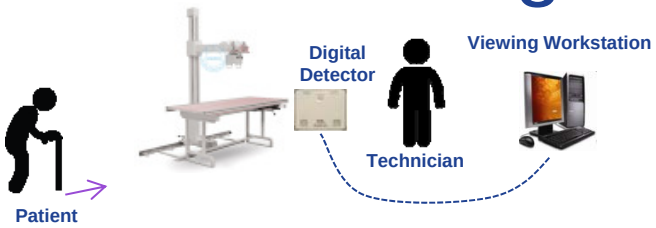
Digital



- Up-front cost, tethered to a room

+ Great IQ, good workflow (preview image), lower operating costs

Portable Digital



- Medium capital cost

+ Good IQ, good workflow (preview image), lowest costs

Best of Both Worlds

A Parable of Requirements Management

Two key requirements

- Digital product interacts with room, but was ‘tethered’, so the new system had to robustly adapt to the new system as it was carried from room to room
  - Response was to store firmware and configuration data in the stationary equipment and download to mobile system each exam
- Digital product had a very fast exam time…the team assumed the same design would be sustained

What happened?

- Download took 30 seconds of the 15 second exam time…discovered ‘too late’
- Yes, **full** two way trace would have prevented it, but is there a simpler way?

Need a simple way to identify interactions between ‘peer’ requirements

What is a use case?

A use case is a story about how a set of behaviors achieves a goal (or meets a need) of a user of the system (actor)

Simple content

- Persona (User or stakeholder)
- Preconditions:
- Postconditions:
- Steps:

Even simpler content is just a picture and the steps

More advanced Content

- Primary Actor:
- Scope:
- Level:
- Stakeholders and Interests:
- Precondition:.
- Minimal guarantee:
- Success guarantee:
- Main success scenario:
- Extensions:

From Writing Effective Use Cases, by Alistair Cockburn, Addison-Wesley Professional; 1st edition (2000)

Actually words (story) can work better than diagrams.

A simple case study (part 1)

Existing Product: GE Digital RAD

Conventional Req’ts Development

- Identify the user needs
- Flow down to the system level requirements
- Perform design reviews of the requirements and decomposition
- Flow down to the subsystems requirements
- Trace to verification and s/s requirements
- Perform design reviews

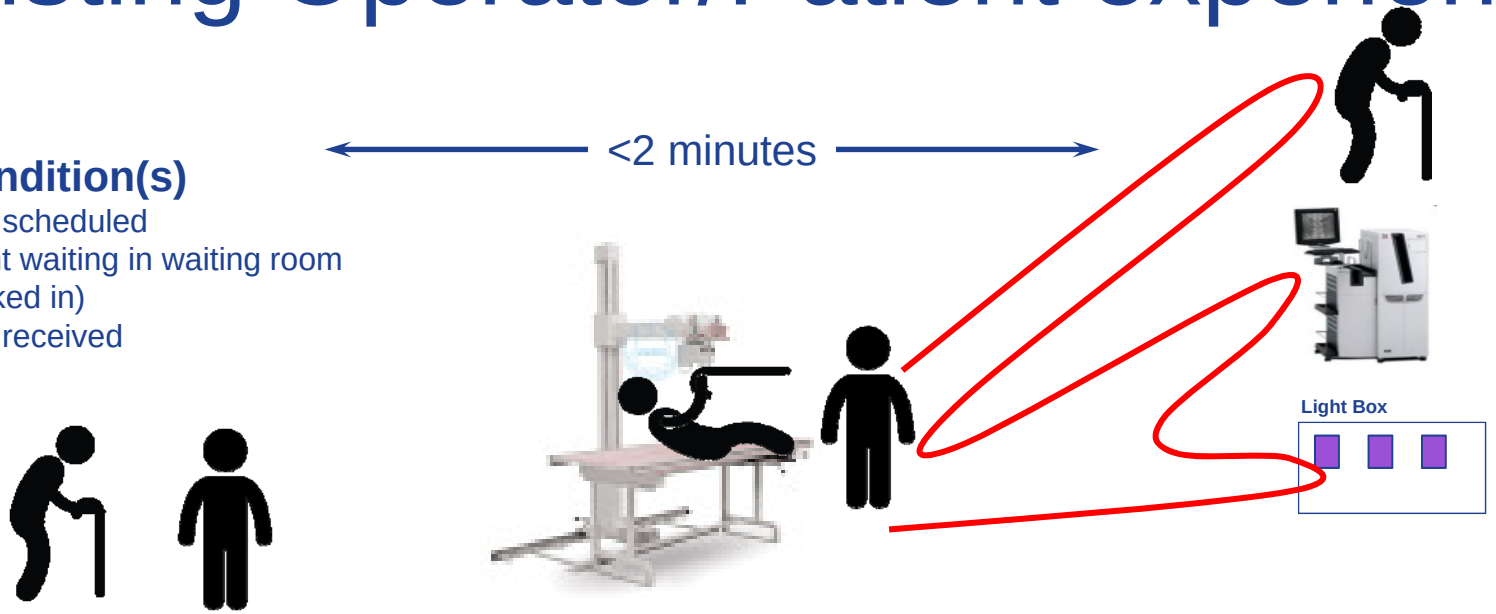
Key elements of the clinical environment

- Tethered to the room
- Hardwired exposure control, hardwired data connection

Key expectations from the user

- Exam time ~1-2 minutes (minimal patient waiting time)
- Results available to physician electronically (immediately)

Existing Operator/Patient experience



Precondition(s)
- Exam scheduled
- Patient waiting in waiting room (checked in)
- Order received

1. Initiate Exam
- Download and review exam
- Find patient and bring to room
- Explain procedure

2. Conduct Exam
- Load film
- Position patient
- Leave room
- Take exam
- Unload film/load 2nd film
- Reposition patient
- Leave room
- Take exam

3. End Exam
- Debrief patient and direct back to waiting room
- Take films to developers
- Develop films
- Take films to light box
- Check positioning…if bad, panic
- Put films in review queue
- Return to room

Key expectations of the operator and the patient
- Exam time 2 minutes (minimal patient waiting time)
- Lots of walking and waiting for operator
- Results available to physician physically

Postcondition(s)
- Exam complete/QA Check complete
- Patient released from exam room
- Films in review queue

A simple case study (part 2)

New Product: Looks the same, but no hidden wires

Goals in developing the new product

- Add flexibility by removing hardwired connection to room infrastructure… enables reusing the detector across rooms
- Simplified workflow compared to ‘analog’ given preview image and no chemicals
- Lower costs compared to digital due to reuse of detector and lower cost detector
- But…needs a simple way for digital detector to adapt to multiple rooms

Design engineer’s expectations

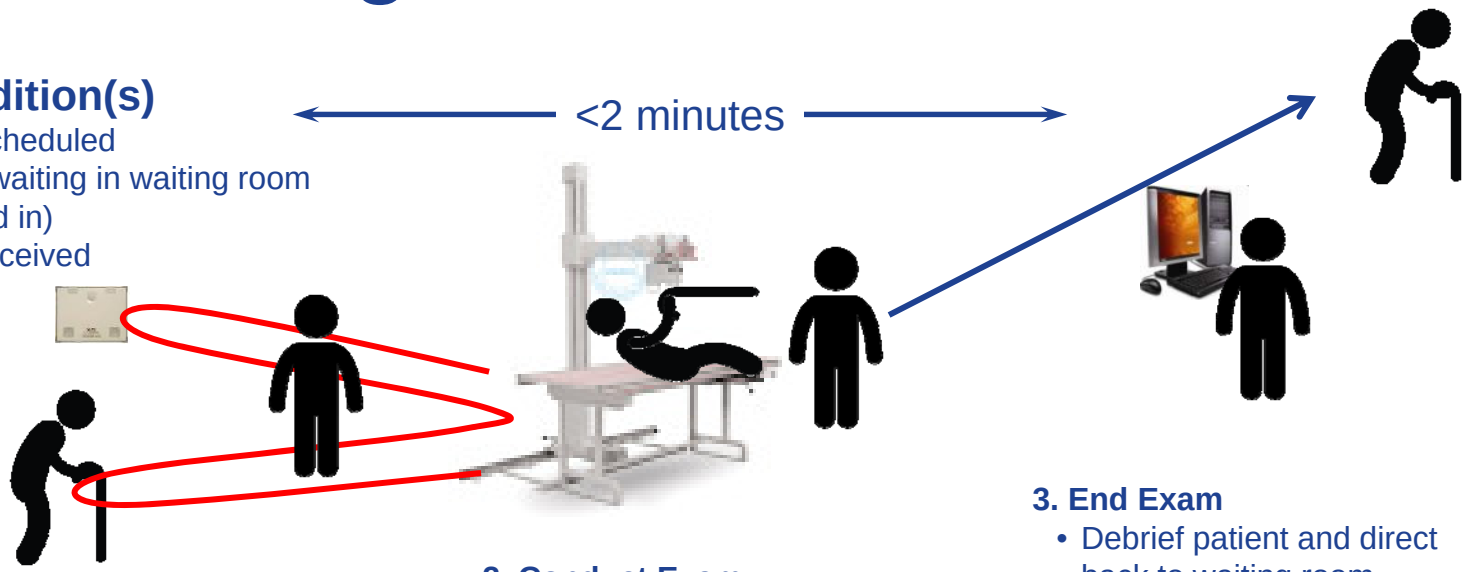
- If keep all the req’ts the same …
- But replace hard network connection with wireless network connection …
- Then user characteristics will be the same.
- But the system will be more flexible because not tied to specific rooms …
- And hospital will benefit because of less exam specific infrastructure in the room

What changed

Precondition(s)

- Exam scheduled
- Patient waiting in waiting room (checked in)
- Order received

<2 minutes



1. Initiate Exam

- Download and review exam
- Find patient and bring to room
- Explain procedure
- **Find detector (if not already in the room), and position in cassette holder)**

- **Detector adjusts to room: ~3 minutes**

This change wasn't handled well

2. Conduct Exam

- Load film
- Position patient
- Leave room
- Take exam
- **QA Check IQ**
- Unload film/load 2nd film
- Reposition patient
- Leave room
- Take exam
- **QA Check IQ**

3. End Exam

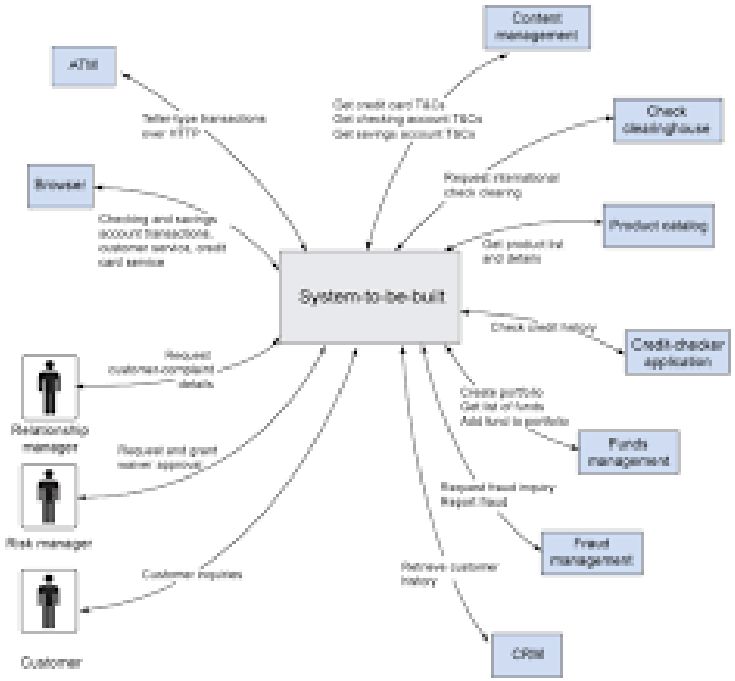
- Debrief patient and direct back to waiting room
- **Network images to review station, notify radiologist**
- Take films to developers
- Develop films
- Take films to light box
- Check positioning…if bad, panic
- Put films in review queue
- Return to room

Postcondition(s)

- Exam complete/QA Check complete
- Patient released from exam room
- **Data networked to server, doctor notified**

Two Additional Tools:
Context Diagrams and Personas

Context Diagrams (BlackBox Analysis): Identifying the Boundaries



How do you identify the users & their needs?

- Draw a diagram where the 'system' being designed is a box
- (Black box == can't see any internal details)
- Identify everything that flows across that boundary and what/who is on the other end

Use those interactions to identify

- Actors (users/stakeholders)
- What they are looking for
- Possible stories (use cases)

Context Diagrams help capture the "Story" behind the Stakeholders' Needs

Simple Baby Warmer example: Defining the Tiers/Configurations

Omnibed

Incubator

Warmer

Omnibed

Nurse

User Interface

- Alarm management
- Display
- User input

Field Engineer

System diagnostics

Room

- Power & Power management
- Light, Heat, Humidity, Acoustics
- Support

Patient weight

Patient

Patient support & enclosure

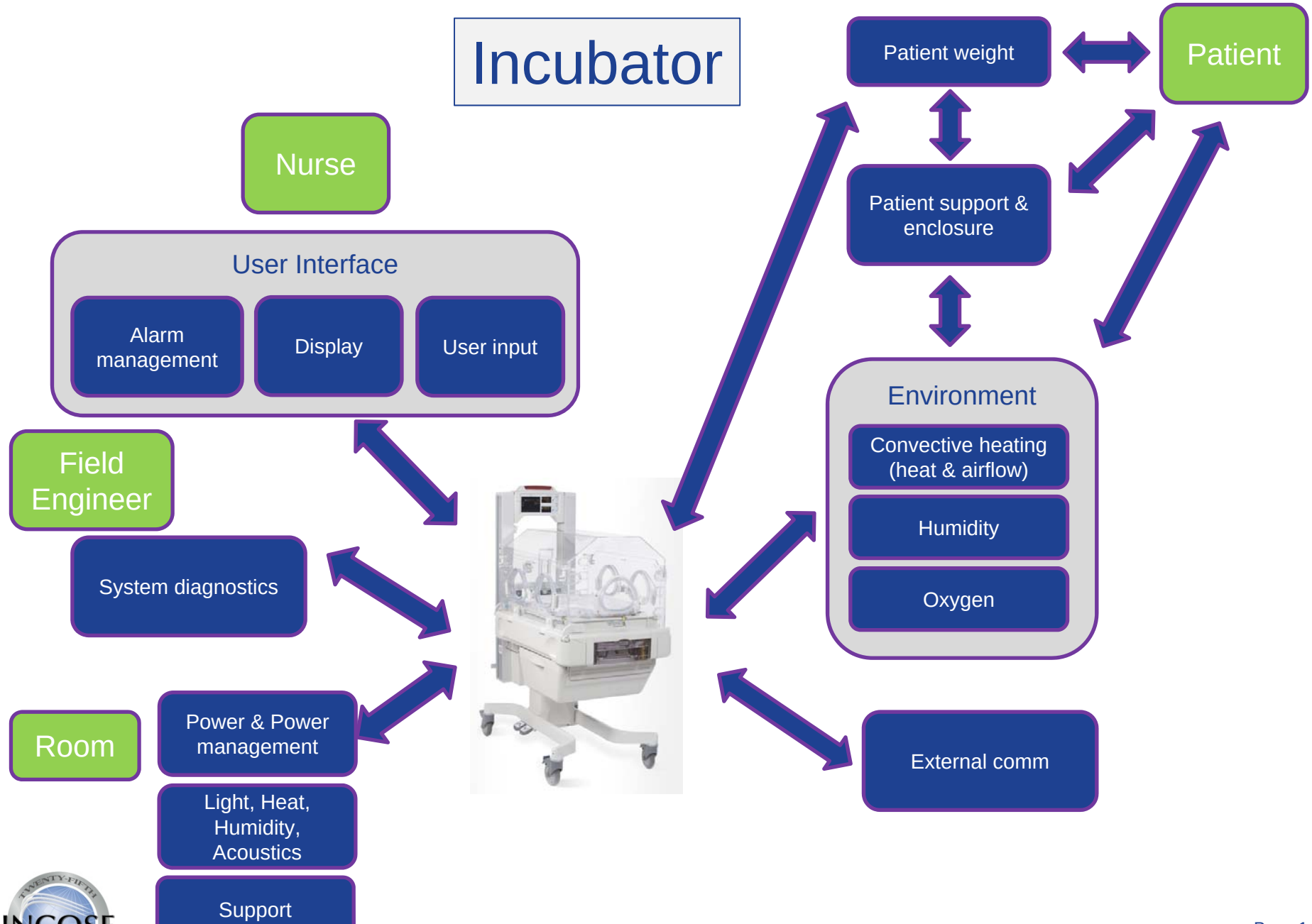
Closed Environment Open

- Convective heating (heat & airflow)
- Humidity
- Oxygen
- Radiant heat

OR

External comm

Incubator



Warmer

Nurse

Lighting

Patient

Patient weight

Patient support & enclosure

User Interface

Alarm management

Display

User input

Environment

Radiant heat

Field Engineer

System diagnostics

SpO_2

Room

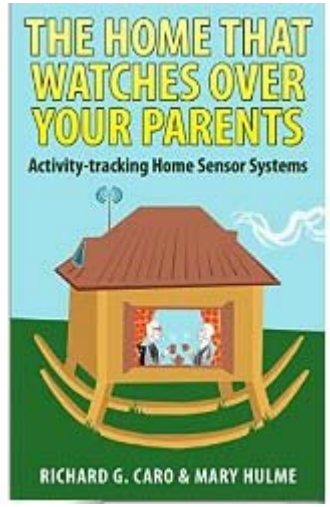
Power & Power management

Light, Heat, Humidity, Acoustics

Support

External comm

Personas: Capturing the "Story" behind the Stakeholders



- The need: Ability to 'monitor' the health of elderly remotely
- Enabled by: new sensors, connectivity, algorithms…
- The Challenge: 'the elderly' and their caretakers are all different
- Reference: "The Home that Watches Over Your Parents"
  - Richard Caro and Mary Hulme, Commonwealth Club, Jan 8, 2015, http://www.commonwealthclub.org/events/2015-01-08/home-watches-over-your-parents
- Defined five "personas"…talk focused on "Go-go" and "No-Go"
  - "Go-Go" is healthy, asymptomatic, and maybe a bit worried about privacy
  - "No-Go" is infirm, and may not 'remember' their concerns
- Book studies 15 different devices
  - Surprisingly, 'visual appeal' turned out to be critical to Go-Go acceptance
  - Actually, care-takers and loved ones had personas also!

Personas help you "Know" your Stakeholders

What might be other examples?

The Microsoft Zune was an MP3 player…but iXxxx + iTunes is a system

- you can get and play any music anywhere on any device…$0.99 per track, cloud storage,

Shazam! Is a way to identify the music you hear…and buy it

SHAZAM®

The kick activated rear door is a way to store your groceries in your SUV

A use case is a complete story in how a user captures value in their context

Assessment

Not all requirements can be elucidated easily via top-down decomposition that is contained completely within the device

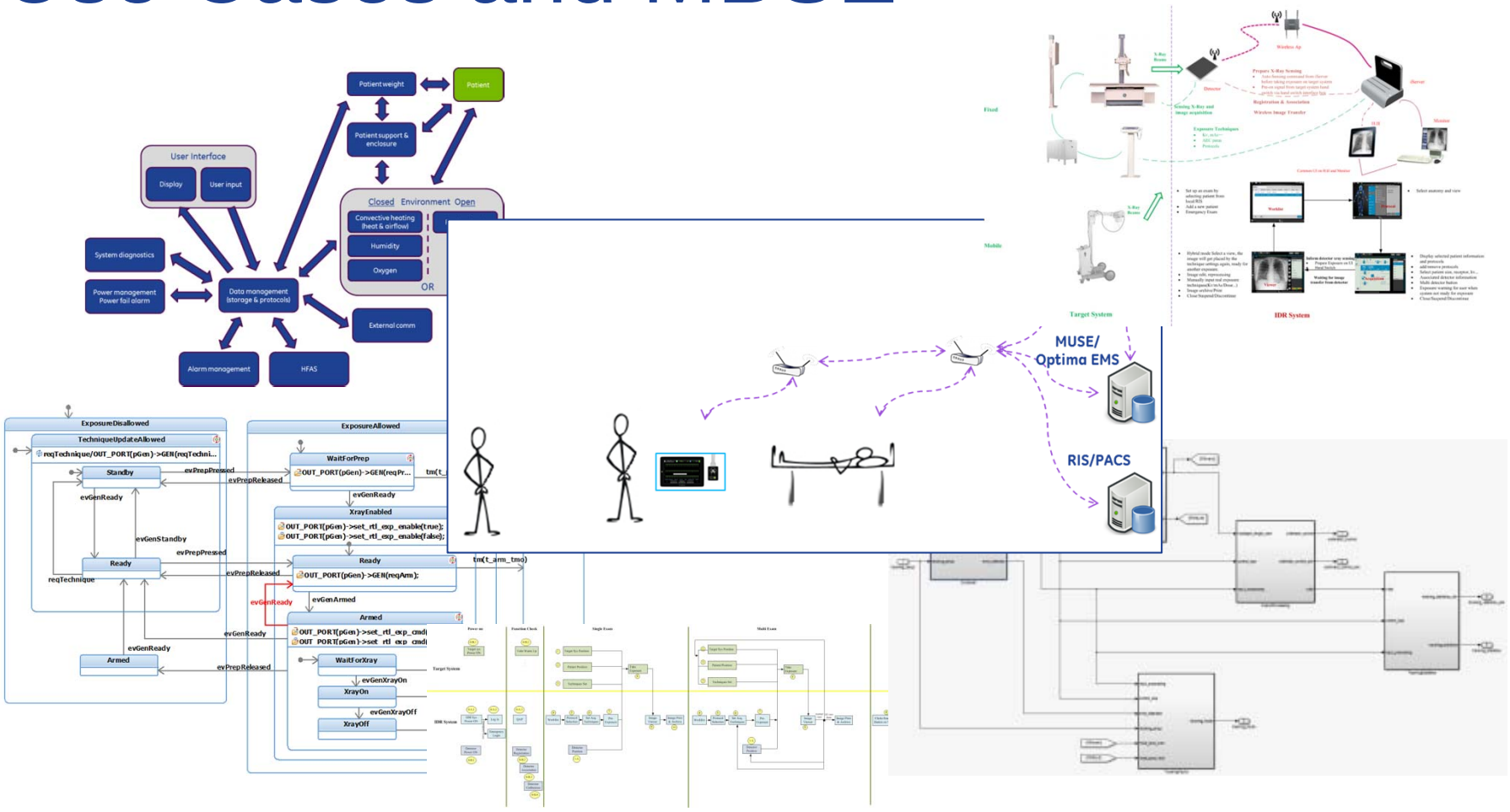
- Designer had assumed that if specs for all components remained the same, then behavior and user experience would be the same
- But some of the new design strategies impacted old requirements flowdowns in unexpected ways

The only way to understand system behavior (and user + patient experience) is to analyze system behavior in the context of the clinical environment

- Use cases can capture user and patient experience
- Context diagrams can capture interactions

The more complex the system, the more likely that behavior will "emerge" once system is allowed to interact with the clinical environment

Use Cases and MBSE



MBSE provides a framework for developing use cases and linking use case analyses to requirements and system specs

Summary – Benefits to Industry of MBSE

Improved Systems Thinking

- Use Case / Behavioral Analysis critical for a complete design specification.
- Logical model to provide high level of abstraction for ease of understanding, improved reuse or design sharing

Improved Communication and Quality

- Visual vs. Textual leads to clearer, more precise communication & better reviews
- Visual designs & models are easier for global teams (less language barrier)
- Verify correctness earlier: model testing vs. just reviews
- Improved design of test cases, derived from weaknesses exposed in the model

Improved Predictability and Efficiency (Time to Market)

- Defects prevented and residual defects found earlier
- Improved leveling of requirements (efficiency in verification and documentation)
- Auto code generation (no translation errors in implementation)

Questions?

Appendix

How to capture use cases?

1. Name the system scope and boundaries.
2. Brainstorm and list the primary actors.
3. Brainstorm and exhaustively list user goals for the system.
4. Capture the outermost summary use cases to see who really cares.
5. Reconsider and revise the summary use cases. Add, subtract, or merge goals.
6. Select one use case to expand.
7. Capture stakeholders and interests, preconditions and guarantees.
8. Write the main success scenario (MSS).
9. Brainstorm and exhaustively list the extension conditions.
10. Write the extension-handling steps.
11. Extract complex flows to sub use cases; merge trivial sub use cases.
12. Readjust the set: add, subtract, merge, as needed.

© Writing Effective Use Cases, by Alistair Cockburn,
Addison-Wesley Professional; 1st edition (2000)

Two approaches to requirements management (simplified caricatures)

"Traditional Systems/Waterfall"

- Three 'special' levels of requirements: User, Systems, Subsystems.
  - The role of systems is to capture user requirements/needs, turn them into clear systems requirements and decompose those into clear, executable, and testable subsystem requirements
- Or, create a two way requirements trace (implicitly with many possible "levels" of requirements)
- Often used in a 'contract' environment, where changing requirements is 'heavy'

Agile

- Goals is speed, and maximizing value (customer/market/business)
- Values dialog and working code over documentation (writing lots of requirements)
- Decomposes "epics" (value streams) into use cases and user stories
- Requirements are changed over time as understanding improves

What's Common

- Goal of Maximizing value
- Common focus on clarity (under test driven development, the requirement isn't done till you have a test case written)

What's Different?

- Agile has a goal of speed, flexibility, and learning/adapting
- Waterfall has a goal of efficiency (minimizing rework) and predictability

Possible harmonized approach

User requirements/needs

→ Use Cases

System requirements (systems)

Subsystems requirements

Use Cases (short, ~1 page) description of how a set of systems requirements (~10?) creates value for the customer

How to manage Timing (SE wants Systems Requirements frozen up front, Agile wants minimal documentation up front and jump to working code)

- Understand lead times
- Prioritize "TBDs" based on risk level and lead times (you could integrate lead times into risk)

Use cases are a tool for prioritizing and phasing an incremental approach to requirements

GEHC Modeling Section

GEHC Approach to New Product Introduction

Tradition NPI process

Program Kickoff | System Req'ts Freeze | Hardware Freeze | Verification Complete | Pilot Release | Full Production | Customer Satisfaction

Traditional artifacts

Requirements = DOORs/Trace (text based)

Systems diagrams in "Visio" (FBD, state machines, activity diagrams, …)

"Quantitative" performance simulations

Challenges

- Lack of customer focus
- Scope creep
- Late integration issues
- Lack of model integration
- Poor requirements leveling (capturing design as reqts)

Recent additions

- Formal Reliability process & team
- Formal Usability process
- Agile methodology (for SW)
- Design for Producibility
- Design for Six Sigma (revitalization)

How Modelling fits in

Systems

- Physics (IQ)

Systems

- *Behavioral*
- *Customer FoM model*

HW: Performance Models

- EE: Cadence/Mentor (Chip->Board)
- ME: Thermal, Structural, Acoustic/Vibration, Life
- Reliability allocations and models
- Should cost modelling

SW: UML models

MFG: Capacity/Cost Models

- Scrap/Cost models
- Capacity/workflow models

The industry faces many challenges

The medical industry product developers face problems with ……

- Extreme time to market pressures
  - 1st to market usually gains 80% of that market
- Compliance with regulations
  - FDA, IEC, ISO, HIPAA, ICD-10, ACA, etc.
- Defects are VERY costly to handle
  - Want to avoid audit, decrees, warning letters, recalls, etc...
- Most products are developed in a geographically distributed way
  - Need to communicate and define tasks
- Technology is impacting development and delivery
  - IoT, product variants, Mobile Medical Apps, complex deployment models, cloud