Health apps: regulation and quality control

Wednesday 19 November 2014, Academy of Medical Sciences
Developing the first CE-marked medical app: Mersey Burns

Rowan Pritchard Jones
Department of Burns and Plastic Surgery, Whiston Hospital

Available from: https://prezi.com/boyb9fhskpno/academy-med-sci/
The first Mobile Prescription Therapy

BlueStar is the only product FDA cleared for real-time patient coaching and clinical decision support.

- First-in-Class Therapy
- Clinical Outcomes: 2 POINT ↓ A1C
- Reducing Costly Hospital Visits: ↓ 58%
- Prescribed: Rx
- Reimbursed: NDC #89129-0100-01 NCPDP Approved

Patented Clinical & Behavioral Engine

*In earlier versions of BlueStar
Digital Health Landscape

- Data Sharing
- Care Delivery
- Wellness
- Chronic Disease Management

- Minimal to Moderate Regulation
- Not Regulated
- Minimal to Moderate Regulation
- Not Regulated
Managing Chronic Disease Can Be Daunting

Healthcare
Four, 15-minute doctor visits a year

Therapeutics
Average of 9 prescriptions yearly

Self-Management
48 minutes a managing disease(s)

A Small Thing Called “Life”
8,766 hours a year balancing life and a chronic disease
Raw Data Isn’t Necessarily the Answer
What if We Could Transform the Data?

To Deliver Actionable Knowledge...
Diabetes Care, Anytime Anywhere

24/7, automated, real-time guidance and education.

Optimizing Clinical Decisions

Treatment and quality recommendations based on evidence-based guidelines for healthcare providers.
Seamless, End-to-end Deployment*

Provider In-Servicing
Face-to-face physician detailing.

Patient Training
Patients are trained face-to-face or remotely.

Customer Care
Product support for patients & providers.
ADA Recognized as New Type 2 Medication

Source: ADA www.diabetes.org
# Potential Regulatory Framework for mHealth

<table>
<thead>
<tr>
<th>Layer 1: Users</th>
<th>Stakeholders who use the system throughout the solution’s lifecycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 2: Application</td>
<td>The feature set and attributes of the software solution that is deployed</td>
</tr>
<tr>
<td>Layer 3: Environment</td>
<td>The physical, regulatory, and security elements of both the mobile and EHR software</td>
</tr>
<tr>
<td>Layer 4: Devices</td>
<td>The end user mobile Internet devices or hardware that are being used (e.g., cellphones, computers) to deliver the MIT solution and their unique attributes</td>
</tr>
<tr>
<td>Layer 5: Network connectivity</td>
<td>The properties of the interfaces that must be considered to ensure proper persistence, resilience, and availability to support integration</td>
</tr>
<tr>
<td>Layer 6: Services</td>
<td>Awareness, education, and training required to ensure maximum value to all users</td>
</tr>
<tr>
<td>Layer 7: Core integration</td>
<td>The data standards, data mapping, and application/systems/workflow integration</td>
</tr>
<tr>
<td>Layer 8: Operating model</td>
<td>The operating and business aspects of the project, including industry observations, cross-enterprise collaboration, and open innovation</td>
</tr>
</tbody>
</table>
Regulation and oversight for developing embedded medical software

Academy of Medical Sciences/Royal Academy of Engineering Joint meeting on ‘Health apps: regulation and quality control’

Dr Chris Elliott FREng
Leman Micro Devices SA
Mobile devices that sense Vital Signs
Case study

Smartphone Vital Signs System:

- Hardware is small enough and cheap enough to incorporate in a smartphone

- SVSS includes everything for medically accurate measurements of all 5 physiological “Vital Signs”
  - Blood pressure
  - Temperature
  - Respiration rate
  - Heart rate
  - Blood oxygen
  and also ECG

- System includes hardware, app and server

- We believe that our package of technological solutions means that the Smartphone is not a medical device
Issues for the app

• Classification
  – Class IIA in Europe
  – Class II in USA

• Safety
  – IEC Class B: Non-serious injury is possible
  – FDA “Moderate”: failure or latent design flaw could directly result in minor injury, including through incorrect or delayed information

• Design and development
  – risk-based
  – good software practice – planned, through life, structured design, traceable (inc SOUP and OS), verified, documented
  – evolutionary development within that framework

• Usability
  – key issue for consumer devices (and Regulators)

• Security
  – confidential health data
Issues for the system including app

• Security
  – how to avoid pirate app (key issue for Regulators)
  – how to deal with rooted ‘phone
• Function
  – Smartphone must deliver as spec through life
    • software upgrades?
    • other apps? (legitimate 3rd party, pirate, interfering)
• Vigilance
  – fundamental requirement of medical device regulation - monitor performance in use and
    deal with “adverse events” (“arisings” in aerospace language)
  – easier with an internet-connected device and dedicated Remote Server
Some key relevant standards

• Safety
  – IEC60601-3 Section 14 – Programmable Electrical Medical Systems
  – IEC 62304
  – FDA Guidance May 11, 2005

• Usability
  – IEC 62366
  – FDA Guidance June 22, 2011

• Security
  – ISO 27001
  – FDA Guidance June 14, 2013

• Risk Assessment
  – ISO 14971

• Performance
  – ISO81060, ISO 80601-2-61, ISO 80601-2-56/ASTM 1965 ....

All within a QMS framework of ISO13485
Conclusions

• It is possible to develop embedded software to satisfy medical device regulations within a Smartphone environment
• At its heart, it’s just good software engineering but
• Must take account of special circumstances:
  – users are consumers not experts
  – environment must be stable (pirates, hackers, upgrades …)
  – objective evidence is needed to support claims for safety and efficacy to satisfy Regulators before sale
Health apps: regulation and quality control
A nuclear perspective

Professor Robin E Bloomfield FREng
19th Nov 2014
Background

- Trustworthiness of computer based system
  - Safety, security and socio-technical perspective

- Deep and broad experience in nuclear industry
  - Practitioner and researcher; Nusac

- In medical sector
  - Health foundation projects
  - Assessing devices
    - Safety, investment
  - Training manufacturers
  - Research liaison with FDA
Background

• Review of 2005-09 Medical Device Reports (MDRs) found:
  • 56,000 MDRs related to infusion pumps, 710 deaths, 87 recalls
  • Several design problems (e.g. software, user interface) seen as “preventable”
  • FDA concluded that there are “numerous systemic problems with device design, manufacturing, and adverse event reporting”

• FDA decided to **proactively** and **systematically** address the root causes of infusion pump recalls
Nuclear power plant

A generator, driven by the turbine.

6 times the size of a Porsche 918, 8205 x power
Overview

• How critical are the systems
  • What are safety properties? hazards?

• What is framework for assurance
  • Safety Assessment Principles
  • Claims, Argument Evidence

• What is the approach
  • Understanding
  • Excellence of production
    – Compensation
  • Confidence Building
    – Statistical testing, static analysis
Protection systems

Plant signals

Pressure

Temperature

© ADELARD
The need for trust in computer-based systems.

Computer systems play a key role in all layers of defence in depth
• Normal operation - control, control room information
• Limitation and warning systems
• Trip systems
• Post trip shut down
• Severe accident management
“Carrot” diagram

- Intolerable risk level: Risk cannot be justified on any grounds
- The ALARP region:
  - Tolerable only if risk reduction is impracticable or if its cost is grossly disproportionate to the improvement gained
- Broadly acceptable region:
  - Tolerable if cost of reduction would exceed the improvement gained
- Negligible risk level:
  - No need for detailed working to demonstrate ALARP
UK - Safety cases: regulatory obligation

- Safety cases are required by licence conditions.
- The Conditions are *non-prescriptive* and set goals that the *licensee is responsible* for meeting.
- A "safety case" is defined as
  - the document or documents produced by the licensee documentation to justify safety during the design, construction, manufacture, commissioning, operation and decommissioning phases of the installation.
- Safety Assessment Principles (SAPs) describe the safety case process and principles to be covered.
## Safety Assessment Principles for Nuclear Facilities - 2006

<table>
<thead>
<tr>
<th>Fundamental principles</th>
<th>Safety assessment</th>
<th>FP.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>The dutyholder must demonstrate effective understanding of the hazards and their control for a nuclear site or facility through a comprehensive and systematic process of safety assessment.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© ADELARD
For software – ESS 27

<table>
<thead>
<tr>
<th>Engineering principles: safety systems</th>
<th>Computer-based safety systems</th>
<th>ESS.27</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where the system reliability is significantly dependent upon the performance of computer software, the establishment of and compliance with appropriate standards and practices throughout the software development life-cycle should be made, commensurate with the level of reliability required, by a demonstration of ‘production excellence’ and ‘confidence-building’ measures.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Summary of principles

1. Effective understanding of the hazards and their control should be demonstrated
2. Intended and unintended behaviour of the technology should be understood
3. Multiple and complex interactions between the technical and human systems to create adverse consequences should be recognised.
4. Active challenge should be part of decision making throughout the organisation.
5. Lessons learned from internal and external sources should be incorporated
6. Justification should be logical, coherent, traceable, accessible, repeatable with a rigour commensurate with the degree of trust required of the system

Derived from IAEA, UK Principles – EU Harmonics project
### Defence in depth Wenra guidance

<table>
<thead>
<tr>
<th>DiD level</th>
<th>Associated plant condition categories</th>
<th>Objective</th>
<th>Essential means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1</td>
<td>Normal operation</td>
<td>Prevention of abnormal operation and failures</td>
<td>Conservative design and high quality in construction and operation, control of main plant parameters inside defined limits</td>
</tr>
<tr>
<td>Level 2</td>
<td>Anticipated operational occurrences</td>
<td>Control of abnormal operation and failures</td>
<td>Control and limiting systems and other surveillance features</td>
</tr>
<tr>
<td>Level 3.a</td>
<td>Postulated single initiating events</td>
<td>Control of accident to limit radiological releases and prevent escalation to core melt conditions</td>
<td>Reactor protection system, safety systems, accident procedures</td>
</tr>
<tr>
<td>Level 3.b</td>
<td>Postulated multiple failure events</td>
<td></td>
<td>Additional safety features, accident procedures</td>
</tr>
<tr>
<td>Level 4</td>
<td>Postulated core melt accidents (short and long term)</td>
<td>Control of accidents with core melt to limit off-site releases</td>
<td>Complementary safety features to mitigate core melt, Management of accidents with core melt (severe accidents)</td>
</tr>
<tr>
<td>Level 5</td>
<td>–</td>
<td>Mitigation of radiological consequences of significant releases of radioactive material</td>
<td>Off-site emergency response, Intervention levels</td>
</tr>
</tbody>
</table>
Product-based approaches

- Focus on directly showing the desired behaviour, property or reliability
- They can be applied even when standards compliance cannot be shown
- Linked with specific claims about the product or system
- May use claim-argument-evidence structure
The role of CAE – communication and reasoning

- A method for reasoning about dependability (safety, security, reliability, resilience ...) properties of the system
- Communication is an essential function of the case, from this we can build confidence
  - boundary objects that record the shared understanding between the different stakeholders
Justification approaches – the strategy triangle

- Property-based approach
- Safety justification
- Standards compliance
- Vulnerability assessment
# Techniques to address properties

<table>
<thead>
<tr>
<th>PROPERTIES</th>
<th>ANALYSIS TECHNIQUES</th>
<th>TESTING TECHNIQUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functionality</td>
<td>Code review / walkthrough</td>
<td>Regression testing</td>
</tr>
<tr>
<td></td>
<td>Traceability (requirements to code)</td>
<td>Statistical testing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Black box functional testing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Negative testing</td>
</tr>
<tr>
<td>Time response</td>
<td>Design inspection</td>
<td>Black box functional testing</td>
</tr>
<tr>
<td></td>
<td>Worst-case execution time analysis</td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>Numerical analysis</td>
<td>Black box functional testing</td>
</tr>
<tr>
<td>Reliability</td>
<td>Analysis of field data</td>
<td>Statistical testing</td>
</tr>
<tr>
<td>Robustness</td>
<td></td>
<td>Negative testing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fault injection testing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stress testing</td>
</tr>
<tr>
<td>Failure integrity</td>
<td>Failure integrity analysis</td>
<td>Fault injection testing</td>
</tr>
<tr>
<td>Operability</td>
<td></td>
<td>Usability testing</td>
</tr>
<tr>
<td>Security</td>
<td>Security analysis</td>
<td>Security testing</td>
</tr>
</tbody>
</table>
# Techniques to address vulnerabilities

<table>
<thead>
<tr>
<th>VULNERABILITY</th>
<th>ANALYSIS TECHNIQUES</th>
<th>TESTING TECHNIQUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unintended inter-component interaction</td>
<td>Resource usage analysis</td>
<td>Black box functional testing</td>
</tr>
<tr>
<td>Incorrect inter-component interaction</td>
<td>Concurrency analysis</td>
<td>Black box functional testing</td>
</tr>
<tr>
<td>Application code errors</td>
<td>Coding standards compliance</td>
<td>Black box functional testing</td>
</tr>
<tr>
<td></td>
<td>Control/data flow analysis</td>
<td>Unit testing</td>
</tr>
<tr>
<td></td>
<td>Resource usage analysis</td>
<td>Integration testing</td>
</tr>
<tr>
<td></td>
<td>Run-time exception analysis</td>
<td>Random testing / fuzz testing</td>
</tr>
<tr>
<td></td>
<td>Abstract interpretation</td>
<td></td>
</tr>
<tr>
<td>Unspecified functionality</td>
<td>Traceability (requirements to code)</td>
<td>Random testing / fuzz testing</td>
</tr>
<tr>
<td>Errors in embedded components</td>
<td>Black box functional testing</td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

• How critical are the systems
  • What are safety properties? hazards?

• What is framework for assurance
  • Safety Assessment Principles
  • Claims, Argument Evidence

• What is the approach
  • Understanding – hazards, interactions
  • Excellence of production
    – Compensation
  • Confidence Building
    – Statistical testing, static analysis
Medical systems

- Tempo
- Heterogeneous systems
- Patient’s own devices
- Accidental systems
- Ad hoc Apps
- Off label
- Local and global
- Multi-stakeholder

Health Foundation Report

Supplement G:
Safety case use within the medical devices industry

Robin Bloomfield, Nick Chozas, George Cleland
Adelard LLP

This is one of a series of supplements to the report: Using safety cases in industry and healthcare: a pragmatic review of the use of safety cases in safety-critical industries – lessons and prerequisites for their application in healthcare.

To access the report and the other supplements, please visit www.health.org.uk/safetycasesreport

Contents

1 Introduction G2
2 Medical devices G2
3 Assurance cases and infusion pumps G6
4 Other developments G13
5 Medical device standards G14
6 Summary G15
7 Glossary G16
8 References G17
Regulation and Oversight for Developing Automotive Software in Europe

Presented at Academy of Medical Sciences, Nov 2014

Dr Michael Ellims
Sybernetic Ltd.
Surprise!

There is some...
Type Approval

*Production* vehicles can be sold and driven if Type Approved.

Type approval has three parts
- Conformity of production
- Approval of test facilities
- Conformance to regulation

Type approval granted by vehicle certification authority e.g. VCA
Conformity of production

Evidence that a manufacturer can produce something that conforms *every time*.

Quality system – ISO 9001 or equivalent

Control plan

  What, why, how, when and who,

Control plans require approval.
Regulation – a lots of it

UNECE (1958 agreement)

– 133 ...to date
– Numerous amendments and corrigendum
– Folded into regulations periodically...

UN GTR (1998 agreement)

– 15 ...to date
Regulation of Software

UNECE 13     Braking, categories M, N and O
UNECE 13-H    Braking, passenger cars
UNECE 79      Steering Equipment
UNGTR 8       Electronic stability control

Note UNECE 13-H also covers ESC !!!
Software Annexes

Special Requirements to be applied to the safety aspects of complex electronic vehicle control systems

UNECE R13 Annex 18
UNECE R13H Annex 8
UNECE R79 Annex 6
UNGTR 8 clause 132 (almost)
The Annex...

<table>
<thead>
<tr>
<th>Clause</th>
<th>Regulation Covers</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Documentation</td>
<td>What is to be presented to certification authority</td>
</tr>
<tr>
<td>3.2</td>
<td>Description of functions</td>
<td>A description... all input and sensed data all output boundaries of functional operation</td>
</tr>
<tr>
<td>3.3</td>
<td>System layout</td>
<td>Inventory of components Functions of the units Interconnections Signal flow and priorities</td>
</tr>
<tr>
<td>3.4</td>
<td>Safety Concept</td>
<td>safe operation under <strong>non-fault</strong> conditions safe operation under <strong>fault</strong> conditions Software Safety Analysis</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Verification of the system</td>
<td>Under non-fault conditions</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Verification of the Safety Concept</td>
<td>Under fault conditions</td>
</tr>
</tbody>
</table>
IEC 61508

Under German law...

– IEC standards are automatically incorporated
  • Implies 61508 should be used for assessing Annexes
– But 61508 doesn’t match automotive practice
  • e.g. prototypes not considered
– IEC 61508 “allows” derived standards

So the automotive industry created ISO 26262
...and ISO 26262

<table>
<thead>
<tr>
<th>Clause</th>
<th>Regulation Covers</th>
<th>Clause</th>
<th>ISO 26262 Covers</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Documentation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td>Description of functions</td>
<td>3: 5.5</td>
<td>Item Definition</td>
</tr>
<tr>
<td>3.3</td>
<td>System layout</td>
<td>3: 5.5</td>
<td>Item Definition</td>
</tr>
<tr>
<td>3.4</td>
<td>Safety Concept</td>
<td>3: 7.5.2</td>
<td>Safety Goals</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3: 8.5.1</td>
<td>Functional Safety Concept</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Software</td>
<td></td>
<td>Part 6</td>
</tr>
<tr>
<td>3.4.4</td>
<td>Analysis (FMEA, FTA etc.)</td>
<td>7.5.1</td>
<td>Hazard analysis</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Verification of the system</td>
<td></td>
<td>missing</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Verification of the Safety Concept</td>
<td>3: 8.5.2</td>
<td>Functional safety concept Verification report</td>
</tr>
</tbody>
</table>
(a) The formal documentation package for the approval, containing the material listed in Section 3

(b) Additional material and analysis data of paragraph 3.4.4., which shall be retained by the manufacturer, but made open for inspection at the time of type approval.
Software 3.4.2.

In respect of software employed in "The System",

– the outline architecture...,
– the design methods and tools ... identified.

The manufacturer shall be prepared, *if required*, to show some evidence of ... realisation of the system logic, ....
Weaknesses

Type approval is in general “whole vehicle”
- tests of a vehicle on a test track
- tests as specified in the standard

Formally:
- regulations do not specify deep dives
- certifying authority does not always do deep-dives

The Annexes are to some extent “bolt-on”
### Some Context...

Data 2013 Fortune 500 and World Bank data

<table>
<thead>
<tr>
<th>Rank</th>
<th>Entity</th>
<th>GDP/Rev</th>
<th>Rank</th>
<th>Entity</th>
<th>GDP/Rev</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Wal-Mart</td>
<td>$476</td>
<td>26</td>
<td>Ford Motor</td>
<td>$147</td>
</tr>
<tr>
<td>2</td>
<td>Royal Dutch Shell</td>
<td>$459</td>
<td>27</td>
<td>General Electric</td>
<td>$146</td>
</tr>
<tr>
<td>8</td>
<td>Volkswagen</td>
<td>$261</td>
<td>61</td>
<td>Nissan Motor</td>
<td>$104</td>
</tr>
<tr>
<td>9</td>
<td>Toyota Motor</td>
<td>$256</td>
<td>68</td>
<td>BMW Group</td>
<td>$101</td>
</tr>
<tr>
<td></td>
<td>Portugal</td>
<td>$220</td>
<td>68</td>
<td>Électricité de France</td>
<td>$100</td>
</tr>
<tr>
<td>13</td>
<td>Samsung Electronics</td>
<td>$208</td>
<td>90</td>
<td>Boeing</td>
<td>$86</td>
</tr>
<tr>
<td>15</td>
<td>Apple</td>
<td>$170</td>
<td>103</td>
<td>Airbus group</td>
<td>$78</td>
</tr>
<tr>
<td>20</td>
<td>Daimler</td>
<td>$156</td>
<td>155</td>
<td>Robert Bosch</td>
<td>$61</td>
</tr>
<tr>
<td>21</td>
<td>General Electric</td>
<td>$156</td>
<td>181</td>
<td>Uruguay</td>
<td>$55</td>
</tr>
</tbody>
</table>
Financial Motivation

Minimum recall cost $50 –
Ford Foci per year (approximate) - 1 million
Typical production run - 4-5 years

$50 * 1 million * 5 years

$250 million
Thank You!

Contact details:
Email: michael.ellims@tesco.net
Mobile: 075 44 68 58 94
Software for Dependable Systems – *Sufficient Evidence*?

Martyn Thomas CBE FREng
Non-executive Director, Health and Safety Executive
Summary
Towards certifiably dependable software

The building blocks for a credible dependability case

• Explicit Claims
• Evidence
• Expertise
Explicit Claims

No software can be equally dependable in all respects and under all conditions of use.

The dependability case should therefore be explicit about the properties that are being claimed, the assumptions that have been made about the environment, and the level of dependability being claimed.

Different properties may be assured to different levels of dependability
Evidence

Concrete and valid evidence should be provided that substantiates the dependability claims.

Evidence + Assumptions ⇒ Properties \(\text{the dependability argument}\)

Testing is essential, but can *almost never* provide adequate evidence on its own.

Evidence from *analysis* will be required.

Evidence of the development process is also needed, e.g. to show that the software in use is the same as that analysed and tested.
Expertise

To develop dependable software, engineers need expertise in software development, in the application domain, and in the broader system context.

Software is only part of the system. It must work dependably with other software, hardware and the users.

Producing adequate evidence is highly demanding and stretches best practice to the limit. Developers must know the best methods and tools and only deviate from them with good reasons (clearly documented).
The full report can be downloaded, free, from

http://sites.nationalacademies.org/cstb/CompletedProjects/CSTB_042247