Software Test Architectures and Advanced Support Environments for IoT

Jon D. Hagar, Consultant, Grand Software Testing

Jon.d.hagar@gmail.com

PNSQ Conference
The Test Architectures and Advanced Support Environments Opportunity

- IoT is Hot
- Test Architectures are poorly defined and understood
- Supporting Test Environments are often weak
- IoT is in part an evolution of embedded systems

But devices want to be winners!
Today’s Topics

• Introduction
  – Definition of Software Test Architecture and Environment
  – Classification of IoT Device Architectures for Software Test

• Classification of IoT Device Environments Needing Test

• IoT Test Architecture and Environment Viewpoints

• Sample Risks Caused by Insufficient Test Architectures and Environments

• Example of Current Large Scale Test Architecture-Environment Solutions

• Future research Needs

• Summary
Defining The Technology Space

Physical Systems (circa 100,000 BC)

Cyber Systems (1950s)

Cyber-Physical Systems

Copyright 2018 Jon D. Hagar – "Software Test Attacks to Break Mobile and Embedded Devices"
What are These IoT Devices

• Embedded – Software contained in “specialized” hardware...
  – Minimal networking-communications in the beginning
  – But now Networked => IoT and Robotics

• Mobile and handheld smart devices—small, held in the hand, highly connected (web, cloud, servers,...)

To Drive

• IoT – Internet of Things are “traditional” and new devices with software and communication added
Definition of Software Test Architecture and Environment

• **Software Test Architecture** is the process(s) and the product(s) of planning, designing, and constructing tests done with supporting test structures
  – Note: supporting test structures include test: tools, environments, documentation, tooling, viewpoints, and analytics

• **Computing Environment** (to support testing) - The overall structure within which a user, computer, or program operates

• **Viewpoint** – In systems engineering, a viewpoint is a partitioning or restriction of concerns in a system

• **Software User** – Typically in software, humans are the only users who interact with the software system, but in IoT, the user of the software is expanded to include
Defining Software Capabilities

• James Whittaker defines 4 fundamental capabilities that all software possesses
  1. Software accepts inputs from its environment
  2. Software produces output and transmits it to its environment
  3. Software stores data internally in one or more data structures
  4. Software performs computations using input or stored data

• To this, I expand and refine based on IoT context:
  – Performance within time while supporting complex communications
  – Testing with specialized hardware
  – Verifying and validating (V&V) qualities: safety, security and privacy
  – Testing to support different environment contexts and constraints
    Early testing      Holistic Testing      Rapid Testing      Field Testing
Last Year’s Test Architecture Conferences

• Defining test architecture problem

• No standards really address architectures

• Examples of architectures and environments may be critical for testing success

IoT maturing chaos
Classification of IoT Device Architectures for Software Test

- IoT differences
  - Hardware
  - Many possible uses/users
  - Billions of products (devices)
  - Software in everything and emergence of AI/Analytics
  - Costs and schedule (really not new, but important)
IoT Test Architecture Classification
Exercise (not in notes)

• Missing or Added Elements?
Architecture Key Points

• Model is not complete

• Many sub areas under “Tooling”

• Much architecture work needed for test environments

• IoT products are out pacing test processes, standards, architectures, and environments
Classification of IoT Device Environments Needing Test

• In IoT test architecture, one critical component will be environments

  – Current IoT project test in “small” isolated environments

  – The World IoT system function in is bigger

Who is responsible for the bigger picture?
IoT Device Environments Mind Map
Environment Mind Map Considerations

• No agreed to allocation of what is IoT

• Much debate over IoT, IIoT, personal devices, cars, robots, etc.

• Test environments still need to be understood and identified within test architectures for products to be successful

• Worse in environments there are many viewpoints which often are not considered during testing
Exercise: What Environments Must We Add?

- Transport systems
  - Large
    - Vehicles personal
    - Vehicles industrial
  - Public
    - Enterprise
    - Government
    - IT and Comm Networks
  - Security and Public Safety
    - Stores traditional
      - Online stores
        - Hospitality
        - Retail
    - Hybrid speciality
    - Healthcare and Science Network
- Middle IoT Sector
- IoT Environments (sw-hw-system-ops)
  - Industrial IoT (IIoT) Sector
    - Internal transport
    - Waste management
    - Management
    - Robotics
    - Material/inventory automation
  - Factories/Production
  - Commercial
    - Large
    - Industrial
    - Government
    - Government
  - Resource automation
  - Energy
  - Personal
    - Home
    - Medical
  - Consumer IoT Sector
<table>
<thead>
<tr>
<th>Environment Example</th>
<th>IoT ViewPoint</th>
<th>Sub Target</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connected Automobile</td>
<td>User (s) - Non Human</td>
<td>Hardware</td>
<td>Controllers, sensors, motors, batteries</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Software</td>
<td>Onboard App, cloud, third party, Operation system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Comm</td>
<td>Vendor network, Wifi end to end comm, long duration trip/drops</td>
</tr>
<tr>
<td></td>
<td></td>
<td>System</td>
<td>Safety, Security, availability, reliability</td>
</tr>
<tr>
<td>Human</td>
<td>First time driver</td>
<td></td>
<td>Security set up, limit usages, non techie</td>
</tr>
<tr>
<td></td>
<td>Basic</td>
<td></td>
<td>Average user, disability, user help files</td>
</tr>
<tr>
<td></td>
<td>Typical</td>
<td></td>
<td>Child, adult, techie</td>
</tr>
<tr>
<td></td>
<td>Advanced</td>
<td></td>
<td>Race Car, expert on snow,</td>
</tr>
<tr>
<td></td>
<td>Bad actor</td>
<td></td>
<td>Hacker, cracker, human using malware</td>
</tr>
<tr>
<td>Dev</td>
<td>Structural tests</td>
<td></td>
<td>Coverage, static analysis</td>
</tr>
<tr>
<td>Tester</td>
<td>Test process, planning, design, techniques, documentation</td>
<td></td>
<td>ISO 29119, ISO 26262</td>
</tr>
<tr>
<td>Management</td>
<td>Information on</td>
<td></td>
<td>Cost, schedule, time to ship</td>
</tr>
<tr>
<td>Ops</td>
<td>Failure management</td>
<td></td>
<td>Help desk, predicitive analysis tires</td>
</tr>
<tr>
<td></td>
<td>Analytics</td>
<td></td>
<td>Machine learning, AI, privacy</td>
</tr>
<tr>
<td>Stakeholder (owner)</td>
<td>Benfit</td>
<td></td>
<td>Information, self-drive</td>
</tr>
<tr>
<td></td>
<td>Resource</td>
<td></td>
<td>Cost, schedule, savings</td>
</tr>
</tbody>
</table>
IoT Test Architecture & Environment Viewpoint Considerations

- A simple environment for an IoT device in a automobile may miss many viewpoints and targets

- Errors and faults are missed

- Security, reliability, performance and many other qualities are likely to be compromised

  - Consider recent car crash – Tesla in California
### Exercise: View Points (not in notes)

<table>
<thead>
<tr>
<th>Device</th>
<th>View Point</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smart Watch</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## IoT Risk Sampling in Test Environments

<table>
<thead>
<tr>
<th>Risk Area Example</th>
<th>Risk Area Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>The reactive and always-on nature of the devices</td>
<td>Lack of realized benefits promised by IoT vendors</td>
</tr>
<tr>
<td>Heterogeneity and diversity at the same time across many systems and devices</td>
<td>Waste (cost or schedule) caused by failures seen in the field</td>
</tr>
<tr>
<td>Power/battery usage limitations</td>
<td>Errors and failures impact happiness and quality of life</td>
</tr>
<tr>
<td>The massively distributed, highly dynamic, and migratory nature of devices</td>
<td>Disruption in society caused by devices (story of traffic lights)</td>
</tr>
<tr>
<td>The need for software fault-tolerant and recovery</td>
<td>Lack of resources (cost and schedule) for test environments</td>
</tr>
<tr>
<td>Fragmentation of the market place (many vendors)</td>
<td>Lack of responsibility for quality across the system or system of systems</td>
</tr>
<tr>
<td>Configuration management of devices to maintain consistency and qualities</td>
<td>Interoperability and integration across devices and sectors</td>
</tr>
<tr>
<td>Current approaches in testing and test architectures do not scale given billions of devices</td>
<td>Software quality characteristics not met, e.g. High availability, reliability, safety, security, usability, and functionality</td>
</tr>
<tr>
<td>Lack of universal product Comm standards</td>
<td>Hardware quality characteristics not met</td>
</tr>
</tbody>
</table>

Copyright 2018 Jon D. Hagar – "Software Test Attacks to Break Mobile and Embedded Devices"
Examples of In-Use Complex Test Architecture-Environments

<table>
<thead>
<tr>
<th>Example of Complex Test Environment</th>
<th>Refer link</th>
<th>Notes and pictures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded test labs of AreoSpace</td>
<td>17</td>
<td>Software, Hardware, and system integration facilities</td>
</tr>
<tr>
<td>Airbus Iron Bird</td>
<td>18</td>
<td>Airplane with all the &quot;parts&quot;, but can not ever fly</td>
</tr>
<tr>
<td>Auto industry high altitude test lab</td>
<td>19</td>
<td>Test the cars in the real world - picture of drap</td>
</tr>
<tr>
<td>Test bed cities and open evaluation platform</td>
<td>20</td>
<td>Real environment, but with what controls and monitor system? How tests and records?</td>
</tr>
<tr>
<td>Chaos engineering</td>
<td>21</td>
<td>Test on live systems. Risk</td>
</tr>
<tr>
<td>Embedded test environments approaches</td>
<td>4, 10</td>
<td>Lab with scopes, software, modeling, switch in and out zero force pic</td>
</tr>
<tr>
<td>Device hardware qualification levels</td>
<td>22</td>
<td>e.g. space qual’d parts</td>
</tr>
</tbody>
</table>

Copyright 2018 Jon D. Hagar – "Software Test Attacks to Break Mobile and Embedded Devices"
Test Environments and Architectures Needed for Each IoT Type

• Test and project planning problem
• Over simplification is a risk
• Under testing a real possibility
• Project will then to “accept” the risk to get to market
• Test Environment and Architectures needed at each level of IoT Architecture
Typical IoT

Architecture Levels
(and example of effort distribution)

Allocation of Test Efforts

World
- APPS-Data
  - Integrator – Meta Data
  - Filter Gateway – data phase filter
    - Propagator – e.g., WiFi, to Net to Server
      - Client – Router/Controller (local)
        - Object – Code
          - Device hardware – Sensor & Actuators – Calibrations

Development Effort
- User Visibility
- Analytics
- Make ready for users

49%
5%
1%
1%
35%
5%
30%
40%
30%
AI/Analytics Enable Many IoT System

- Data modeling analytics, statistical design, AI and deep learning
  - Model based testing leading automation (millions of tests?)

- Designing AI-deep learning
  - Using data from machine learning but watch

  - Neural Net case study: programmers who write more and better comments make fewer errors (not what I expected)

- In statistics we scientifically design experiments and analyze data
  - How to pick training sets?
  - What is important?
  - What is valid?

- Test Analysis => Thinking
Test Implications of Deep Learning Models

Data Analytics Using AI models may be very useful but thinking humans are still needed - Case Study of Early bug Taxonomy

You need Valid Big Data

You Should Test The Learned Model

You Should Verify The Model Over The Life of the Device

Credit: IEEE Sweden 2018
Future IoT Work Opportunities

- Specialized IoT IV&V facilities at the full system/system-of-system for IoT
- Data analytics with real-time testing in the field and self-healing systems
- Government departments tasked with the focus on IoT (National testbed)
- Privacy and security regulations and standards
- Model Based Testing (MBT) and simulation driven test beds
- Predictive Maintenance
- Distribution and heterogeneous systems on the Internet
- Industry and government test labs with independence (cost and schedule)
- Testing to address the fragmentation of IoT
- IoT Test support tools
- IoT data analytics and deep learning
- Support process standards e.g., ISO and IEEE
Summary

• The research for this paper has indicated the need for more work in IoT test architecture and environments
  – An IoT classification example for test architecture and environments
  – Demonstrated viewpoint usage in testing
• Existing and successful test facilities from the embedded software device world
• Some projects will make IoT devices just “good enough” to continue
• The lack of being “good enough” will lead to many IoT projects learning by failures in the field
• Risks outlined in this paper are only part of the IoT problem
• IoT, IIoT and IoT of everything are being rushed, implemented and fielded before many of these test architecture and environment issues are even considered, let alone solved and resources will be wasted
References:  Thank You (ideas used from)

- François Coallier, Presentation on System of Systems Engineering Challenges in IoT based Systems”, IEEE 12th International Conference on Systems of Systems Engineering (SoSE), 2018-06-21
- Jane Fraser, The Internet of Things in Action: Testing Anki’s OVERDRIVE Racing Game, May 2016, STAREast Conference Presentation
- Jon Hagar, Defining the Phrase “Software Test Architecture”, 4th International Workshop on Software Test Architecture, 2018, IEEE
- Steven Latre, Philip Leroux, Tanguy Coenen, “City of things: An integrated and multi-technology testbed for IoT smart city experiments” Published in: Smart Cities Conference (ISCC), 2016 IEEE International
- IEEE 1012 IEEE Standard for System, Software, and Hardware Verification and Validation
- Sokwoo Rhee (sokwoo.rhee@nist.gov), “Cyber-physical systems, IoT, and smart cities “global city teams Challenge”, Embedded system conference 2015
- Marc-Florian Wendland, UML TESTING profile 2 OMG’s new standard for model-based testing, SDL-Forum, Berlin, 12th October, 2015
- Brice Morin, Nicolas Harrand, and Franck Fleurey, Model-Based Software Engineering to Tame the IoT Jungle, IEEE Software, Jan/Feb 2018, P30-36
- Laura Belli, Simone Cirani, Luca Davoli, Andrea Gori, Mirko Mancin, Marco Picone, and Gianluigi Ferrari, University of Parma, “Design and Deployment of an IoT Application-Oriented Testbed”, IEEE COMPUTER 0018-9 162 / 15/2015, p 32-41
- https://nextm2m.com/fragmented-iot-ecosystems-are-destined-for-connectivity-interoperability-and-security-problems/