Architecting an Online Social Resource for Amateur Filmmakers Using a Hybrid Systems Engineering and Agile Approach

Thomas J. Smith II

Advisor: Dr. Galloway

12.09.15
Agenda

- System Overview and Purpose
- Candidate Concepts
- Systems Engineering and Agile
- System Requirements/Agile Stories
- Measures of Effectiveness/Requirements Verification
- Architectural Solution (System Architecture and Workflow)
- Risk Analysis and Ethical Concerns
- Lessons Learned
- Conclusion
System Overview and Purpose: Thomas Background

- Graduated with B.A. in Creative Writing, Film Studies minor in 2008
- LMU Systems Engineering, Computer Science Technical Focus
- Skills background:
  - Technical:
    - Electronics repair (Refurbished Circuits)
    - Media Content Transcoding and Digitization, Digital Inventory Management (Sony)
    - Integration and Workflow Automation, Software Engineering (Sony)
    - Technical Hardware and Software Support (Alchemy)
    - Technical Creative Workflow Solutions (Alchemy)
  - Creative:
    - Screenwriting
    - Novel Writing
    - Poetry
    - Short Film Production
- This project is both a technical and creative challenge
System Overview and Purpose: Independent Film Industry Challenges (The Problem)

- Budget/funding (filmmaking is expensive overall)
  - Particularly: Equipment, talent, locations
  - Finding these things is also difficult
- There are a lot of informative and advantageous resources, but they are:
  - Difficult to find (especially for a specific genre or other specialization)
  - Expensive to participate in
  - Difficult to participate in (they fill up quickly)
System Overview and Purpose: 
Mission

- To ultimately invigorate the independent filmmaking community
  - By promoting engagement and penetration
  - By providing a social element that encourages collaboration, apprenticeship, and mentorship
  - By providing accessible, relevant resources that aren't oversharded (different demographics see different resources)
- To provide a solution for B-Roll content that:
  - Is an inexpensive or free solution
  - Has rapport amongst filmmakers
  - Is an extensive but relevant content library
- To reduce logistical efforts thus allowing filmmakers to focus more time on creative challenges instead
- To become a national authority and resource for independent film
System Overview and Purpose: ConOps

- Intended as a resource primarily for amateur and independent filmmakers in the contiguous United States
- Aims to serve the purpose of cultivating a helpful filmmaking community
- Supplemental goal is to make non-creative filmmaking challenges easier
  - Making the "filmmaking experience" cheaper, or at least allow certain costs to be deferred
  - Acquiring B-Roll on website reduces footage tasks and complexity
- Independent filmmakers can interface with other filmmakers, cast, crew
- Free to join and free to perform the majority of functions
- Architecture priorities (in order): security, performance, cost, schedule
System Overview and Purpose: Related Work

- **Stock Footage Libraries:**
  - Shutterstock.com
  - GettyImages.com
    - A la carte, royalty-free, but expensive
  - VideoBlocks.com
    - Unlimited footage for affordable annual rate, but high-quality content buried
  - VideoHive.com
    - A la carte, royalty-free, and affordable, high-quality

- Bee-Roll.com proposes primarily free services, any premium services or content are affordable
System Overview and Purpose: Related Work

- Social Networking/Production Job Placement:
  - Craigslist.org
    - Runs entire gamut of online classifieds, anyone allowed to post (relevant content buried), "inclusive" domain
  - Facebook.com
    - Recreational focus (but also used for professional use), "inclusive" domain
  - LinkedIn.com
    - Professional focus, "inclusive" domain
  - StaffMeUp.com
    - Professional focus, "exclusive" domain

- Bee-Roll.com proposes an "exclusive" audiovisual media production industry domain catered to the inexperienced and those without the luxury of big studio budgets
Candidate Concepts

- **Methodologies:**
  - Systems Engineering
  - Agile
  - Hybrid Solution: Agile Systems Engineering

- **Workflow Architecture (how content is handled and what will be done):**
  - B-roll/stock footage content library
  - Online social networking

- User-level Cloud Services
- Content Delivery Network (CDN)
Candidate Concepts

- Infrastructure:
  - Hardware:
    - Server Colocation
    - Dedicated Servers
    - Virtual Private Servers (VPSs)
  - Open Source vs. Licensed Software
- Software:
  - Operating System:
    - Windows OS
    - Linux OS
  - Webpage Coding/Client Software:
    - Flash
    - Java
    - Javascript
    - HTML 4
    - HTML5
    - CSS
Candidate Concepts

- Infrastructure (cont'd):
- Software:
  - Backend Coding/Server Software:
    - LAMP
      - Linux
      - Apache
      - MySQL
      - PHP (or Perl or Python)
    - NGINX
    - MongoDB
If a Website Were a House...

**HTML**
House's structure. How many rooms it has, how many stories it is, whether it has a fireplace or not.

**Javascript**
House's functionality. Garage door opens with remote, lights operated with switches, HVAC, articulating doors and sliding windows.

**PHP/Python/Perl**
Construction workers that build the house.

**CSS**
House's style. Room dimensions, where doors and windows are located, textures and colors of everything.
# Systems Engineering vs. Agile

<table>
<thead>
<tr>
<th>Why Systems Engineering?</th>
<th>Why Agile?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driven by robust requirements</td>
<td>Driven by prioritized user stories (a wish list)</td>
</tr>
<tr>
<td>Changes in requirements can prove costly even in early stages</td>
<td>User stories can be changed or removed altogether at any time with minimal consequences</td>
</tr>
<tr>
<td>Lean encouraged, but optional</td>
<td>Lean inherent/expected</td>
</tr>
<tr>
<td>Generally long iterations</td>
<td>Very short iterations</td>
</tr>
<tr>
<td>Emphasizes extreme reliability, sustainability, long term compatibility</td>
<td>Emphasizes extreme flexibility via immediate capability, scalability, feature tradability</td>
</tr>
<tr>
<td>Requirements to be written by systems engineers while conferring with stakeholders</td>
<td>User stories to be written by stakeholders (esp. users), not developers</td>
</tr>
</tbody>
</table>
Systems Engineering + Agile

- Agile does not replace Systems Engineering! It will supplement it.
- Agile will be used underneath the systems engineering wrapper:
  - SE will be used for top-level, rigid performance requirements, and high-level infrastructure: "the chassis"
  - Agile will be used for managing ever-evolving user requirements "under the hood"
- Systems Engineering processes will have phased deployment, delivering every 1-2 years.
- Agile will address less complex, but more urgent issues, delivering at least every 1-2 weeks.
- Agile's user stories are like requirements, but they can be easily reprioritized, added, rewritten, or removed altogether.
  - User stories are supposed to be written by stakeholders, especially users—not developers.
  - Users decide what the product has and should do, development prioritizes those ideas.
System Requirements:
Top-Level Requirements and Verification(SE)

1. Bee-Roll.com shall respond to ≥70% of client requests for ≥99.00% of the time.

2. Regardless of system load, any given client request shall receive a server response in 300 milliseconds or less for ≥90% of all cases, the remaining cases shall receive a response in ≤3 seconds with the exception of Distributed Denial of Service (DDoS) attack scenarios.

3. Regardless of system load, any given client request shall result in a page load time of ≤1 second for ≥70% of all cases, but ≤10 seconds page load time for the remaining cases with the exception of DDoS attack scenarios.

4. The domain name system (DNS) shall route a user to the server nearest to their geographical location unless that infrastructure is saturated, in which case they will be routed to the nearest available set of servers.
5. All web pages shall be coded in technologies natively supported by the 5 leading web browsers (HTML5, JavaScript, and CSS at this time).

6. Proxy files shall be created from the source files submitted by registered users with a burned-in Bee-Roll.com watermark.

7. Proxy files shall be ≤15 seconds in duration and transcoded to match the bit rate and resolution of the source file.

8. Inappropriate content shall be reviewed within an hour of being flagged by a trusted moderator or employee.
System Requirements: Top-Level Requirements Verification (SE)

1-3. Response Time and Uptime Requirements:

**Analysis:** Performance metrics will be monitored with sample groups to model and verify for entire system

4. Geolocation-Routing Requirement:

**Analysis:** System will be tested at various loads, user geographical locations, and modeled accordingly to verify

5. Native Browser Languages Requirement:

**Inspection:** This coding convention can be visually verified

6, 7. Proxy File Requirements:

**Test:** Test cases with a variety of inputs will be used to verify outputs

8. Inappropriate Content Requirement:

**Demonstration:** Results will be verified with production data with production-scale traffic
System Requirements: Some Initial Agile User Stories

- Users can log in using mainstream social networking accounts
- Registered users can upload video content to share
- Users can update their profiles to reflect film roles
- Producers and Directors can create casting calls and job postings
- Inappropriate content can be flagged or reported
- Registered users can message each other
- B-Roll content can be searched for using keywords
- Registered actors can post filmographies and audition videos
- Registered crew can post resumes, filmographies, and reels
- User stories will be verified via Testing (QA Team) and validated
Measures of Effectiveness

1. The number of HTTP requests that receive a server response
2. The time it takes for a server to respond to an HTTP request
3. The time it takes for a page to load on a client device
4. How many users are routed to the server nearest them
5. How many of the web page technologies are supported natively by web browsers
6. The number of proxy files with burned in Bee-Roll.com watermarks
7. The duration and bit rate of the proxy files
8. The time between an item being flagged and someone reviewing it
Measures of Effectiveness

- The number of amateur/aspiring filmmakers aware of Bee-Roll.com
- Total number of users
  - Registered users
  - Registered users that contribute (comments, postings, media)
  - Percentage of unique unregistered visitors per month that do not register
- Total number of projects represented on Bee-Roll.com (project postings)
- Total number of video objects in content library
  - Percentage of video objects with 5 views or more or at least one download
- Duration of downtime during a DDoS attack
System Architecture and Workflow: OV-1
System Architecture and Workflow: System Interface View

Web Infrastructure

Other Virtual Private Servers

Content Delivery Network

Bee-Roll.com Web Infrastructure

Perl, Python, or PHP crunches data

Dynamic Data Req

Data Response

Bee-Roll.com Web Server

Firewall

Data Pass-Thru

Domain Name System

HTTP Request

HTML, CSS, JS

User Device

Client/Web Browser

Renderers and Interpreter

DB Server

Data Response
System Architecture and Workflow: Lean

- For this context, the site's content and the site itself will be considered products.
- The tools and processes detailed in this architecture intended to be conducive to lean processes:
  - Use of highly-customizable software and hardware platforms—generic, rigid tools tend to be wasteful.
  - Customizable tools can be fine-tuned to eliminate waste.
- Examples:
  - Linux distributions and running processes.
  - Linux CPU/RAM utilization (low) vs. Windows Server (and use of GUI, high). Linux can be customized to run with or without GUI, certain processes, etc.
  - **DRY** (don't repeat yourself): Inefficient code more difficult to maintain, more CPU utilization, using same code for desktop and mobile clients.
System Architecture and Workflow: Lean: Cloud Computing (Elasticity)

- AKA infrastructure as a service (IaaS)
- Building data centers or otherwise purchasing your own servers is wasteful (unless you’re an internet giant)
- No equipment to physically manage (less overhead)
- Infrastructure can be scaled appropriately with reasonably less effort, time, and $$$
- VPSs with predetermined server images are spooled up almost instantly, then taken offline when no longer needed
- Elastic cloud providers only charge for what you use (i.e. AWS)
System Architecture and Workflow: Cloud Computing Elasticity Overview

# Of Active Users

x1000
System Architecture and Workflow: Lean: Automation

- Automation = Lean? Not necessarily
- Will be used for tedious, repetitive, predictable tasks
- Examples:
  - Scaling up server farms
  - Propagating configuration changes
  - Updating multiple servers
  - Running backups
  - Simple user account creation and management (i.e. new website users)
- Tools:
  - Ansible (orchestration tool—use of playbooks)
  - Python (scripting language)
Risk Analysis: Technological Risks

- Infrastructure Saturation
  - Utilization of any infrastructure module beyond capacity
  - Often the result of inaccurate market data, underestimation, or improper configuration

- Security Concerns:
  - Data breaches
  - Forced infrastructure saturation (DDoS attacks)
  - Security exploits

- Total failure or loss of infrastructure
  - Power outages
  - Fires
  - An act of God
  - Cloud service company suffers cyber attack or goes out of business
Risk Analysis:
Non-Technological Risks

- Losing relevance or inability to establish user base in market
  - Market could be dominated by other players
  - Users could just lose interest or not care at all
  - Content library is too small or full of low-quality content
  - Website not intuitive or functional enough, or difficult to find in the first place

- Not having enough capital

- Legal Risks
  - Lawsuits
  - Reckless or malicious user actions
Risk Analysis: Ethics

- Ethical violations against SE code, but are also risks for the business
- Significant ethical concerns revolve around how Bee-Roll.com manages its users
- It is possible for users to upload lewd, degrading, defamatory, offensive, or otherwise inappropriate content
  - Videos
  - Comments, job postings, and user names
- Prospective collaborating filmmakers will inevitably meet in person
  - Danger risk associated with meeting strangers
- Security is paramount
  - Secure storage, processing, and deletion of user data, even after company is gone
  - Secure site, uncompromised by external parties (free from malware, viruses)
# Risk Analysis: 5x5 Matrix

## Pre-Mitigation

<table>
<thead>
<tr>
<th>Likelihood</th>
<th>Consequence</th>
<th>Primary Action Type</th>
<th>Wt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>Transfer</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>Mitigate</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>Accept</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>Accept</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>Accept</td>
<td>6</td>
</tr>
</tbody>
</table>

## Post-Mitigation

<table>
<thead>
<tr>
<th>Likelihood</th>
<th>Consequence</th>
<th>Primary Action Type</th>
<th>Wt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5</td>
<td>Transfer</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>Mitigate</td>
<td>24</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>Accept</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Accept</td>
<td>15</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Accept</td>
<td>6</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Risk</th>
<th>Risk Type</th>
<th>Likelihood/Consequence</th>
<th>Primary Action Type</th>
<th>Wt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Infrastructure Saturation</td>
<td>Technical</td>
<td>4L, 4C</td>
<td>Transfer</td>
<td>16</td>
</tr>
<tr>
<td>2) Security Concerns</td>
<td>Technical</td>
<td>5L, 5C</td>
<td>Mitigate</td>
<td>25</td>
</tr>
<tr>
<td>3) Total failure or loss of infrastructure</td>
<td>Technical</td>
<td>2L, 3C</td>
<td>Transfer</td>
<td>10</td>
</tr>
<tr>
<td>4) Loss of or inability to establish relevance</td>
<td>Social</td>
<td>3L, 2C</td>
<td>Accept</td>
<td>15</td>
</tr>
<tr>
<td>5) Not having enough capital</td>
<td>Financial</td>
<td>2L, 2C</td>
<td>Mitigate</td>
<td>10</td>
</tr>
<tr>
<td>6) Inappropriate user content</td>
<td>Ethical</td>
<td>5L, 4C</td>
<td>Accept</td>
<td>20</td>
</tr>
<tr>
<td>7) Risk of danger to users when networking in person</td>
<td>Ethical</td>
<td>2L, 3C</td>
<td>Accept</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Risk</th>
<th>Risk Type</th>
<th>Likelihood/Consequence</th>
<th>Primary Action Type</th>
<th>Wt.</th>
<th>Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Infrastructure Saturation</td>
<td>Technical</td>
<td>4L, 4C</td>
<td>Transfer</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>2) Security Concerns</td>
<td>Technical</td>
<td>5L, 5C</td>
<td>Mitigate</td>
<td>6</td>
<td>19</td>
</tr>
<tr>
<td>3) Total failure or loss of infrastructure</td>
<td>Technical</td>
<td>2L, 3C</td>
<td>Transfer</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>4) Loss of or inability to establish relevance</td>
<td>Social</td>
<td>3L, 2C</td>
<td>Accept</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>5) Not having enough capital</td>
<td>Financial</td>
<td>2L, 2C</td>
<td>Mitigate</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>6) Inappropriate user content</td>
<td>Ethical</td>
<td>5L, 4C</td>
<td>Accept</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>7) Risk of danger to users when networking in person</td>
<td>Ethical</td>
<td>2L, 3C</td>
<td>Accept</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>
## Risk Analysis: Assessment and Control

<table>
<thead>
<tr>
<th>Risk (Abridged)</th>
<th>Mitigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infrastructure Saturation</td>
<td><strong>Transfer:</strong> Use of an elastic IaaS (the cloud)</td>
</tr>
<tr>
<td>Security Concerns</td>
<td><strong>Mitigate:</strong> Webserver security best practices, data encryption, encrypted link: secure sockets layer (SSL)</td>
</tr>
<tr>
<td>Total failure or loss of infrastructure</td>
<td><strong>Transfer:</strong> Use of geographically disparate infrastructure with redundancies and failovers, multiple IaaS services used</td>
</tr>
<tr>
<td>Loss of or inability to establish relevance</td>
<td><strong>Accept:</strong> Focus groups will be conducted, independent filmmakers interviewed, competition monitored. Problem expanded to help mitigate this risk.</td>
</tr>
<tr>
<td>Not having enough capital</td>
<td><strong>Mitigate:</strong> Use crowdfunding and curb expenditures with lean thinking, use of open source software</td>
</tr>
<tr>
<td>Inappropriate user content</td>
<td><strong>Accept:</strong> This will always happen, but reactive measures can be taken with a moderator (employee or privileged user)</td>
</tr>
<tr>
<td>Risk of danger to users when networking in person</td>
<td><strong>Accept:</strong> Users will be warned with guidelines about meeting in person (i.e. meeting in public places)</td>
</tr>
</tbody>
</table>
Lessons Learned

- Unexpectedly one of the most valuable heuristics: “Sometimes, but not always, the best way to solve a difficult problem is to expand it.”

- There are more players attempting to address this problem (in some form) than originally expected, some are successful.

- In alignment with Agile, SE, and recent software implementation trends, the end user/customer should be heavily involved.

- Establishing the initial user base will be the greatest non-technical challenge, overshadows technical challenges.
Lessons Learned

- SE calls for structure and well-defined requirements, but doesn’t discourage creativity
- SE methodologies can be used in conjunction with other methodologies (i.e. Agile, Spiral Development)
- Lean Thinking is naturally a part of Agile
Conclusion

- Problem expanded from just B-Roll challenges to larger independent film industry issues
- Currently in study period for a little longer, will continue to acquisition phase around February 2016
- Hybrid Systems Engineering and Agile approach yields a process that is robust, which requires swift proactive and reactive flexibility
- Open source software and cloud services are used to mitigate ballooning costs
- Cloud services are also used to meet demanding expectations in as little time as possible
Q&A
Resources


Resources


Image Resources

1. http:// ouralchemy.com
3. http://www.screenmedia.co.uk/media/135946/agile-methodolody_695x260.jpg
4. http://3.bp.blogspot.com/-mUw1WXE7p_k/Uhr_-ITmHil/AAAAAAAIAtA/ZGRhZsWsklw/s1600/firefox-internet-explorer-google-chrome-safari-opera.png
5. http://d1qmdf3vop2l07.cloudfront.net/delightful-starling1.cloudvent.net/compressed/5dfc3eb02db8209f5fb7dbc4dc3c3eb5.png

Thomas J. Smith II 38