Let’s Play Poker: Effort and Software Security Risk Estimation in Software Engineering

Laurie Williams
williams@csc.ncsu.edu

Another vote for…

“Everything should be made as simple as possible, but not simpler.”

--Albert Einstein
Estimation

Planning Poker

How many engineers? How long?

Protection Poker

Effort Estimation: Planning Poker

How many engineers? How long?

Historical Effort Estimation

Gut feel often based on:
- Disaggregation
- Analogy
- Expert opinion

"I would appear hopeless, that your gut feel was only subjective."

Comming up with the plan

<table>
<thead>
<tr>
<th>Desired Feature</th>
<th>Divide by Velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>50 story points</td>
<td>5 story points/iteration</td>
</tr>
<tr>
<td>30 iterations</td>
<td>6 iterations</td>
</tr>
<tr>
<td>June 10</td>
<td></td>
</tr>
</tbody>
</table>
Estimating “dog points”

- Estimate each of the dogs below in dog points, assigning each dog a minimum of 1 dog point and a maximum of 10 dog points.
- A dog point represents the height of a dog at the shoulder.
  - Labrador retriever
  - Terrier
  - Great Dane
  - Poodle
  - Dachshund
  - German shepherd
  - St. Bernard
  - Bulldog

What if?

- Estimate each of the dogs below in dog points, assigning each dog a minimum of 1 dog point and a maximum of 100 dog points.
- A dog point represents the height of a dog at the shoulder.
  - Labrador retriever
  - Terrier
  - Great Dane
  - Poodle
  - Dachshund
  - German shepherd
  - St. Bernard
  - Bulldog

Harder or easier?
More or less accurate?
More or less time consuming?
Estimating story points

• Estimate stories relative to each other
  – Twice as big
  – Half as big
  – Almost but not quite as big
  – A little bit bigger

• Only values:
  – 0, 1, 2, 3, 5, 8, 13, 20, 40, 100

Near term iteration “stories”  A few iterations away “epic”

Diversity of opinion is essential!

Vote based on:
• Disaggregation
• Analogy
• Expert opinion
Not working as fast as planned?

Desired Features

- 30 story points
- 6 iterations
- 5 story points
- 1 iteration
- 10 iterations
- July 8

Estimate Size

Divide by Velocity

3 story points

Iteration/Release plan

Iteration/Release plan

(Subjective) Results of Planning Poker

- Explicit result (<20%):
  - Effort Estimate
- Side effects/implicit results (80%+):
  - Greater understanding of requirement
  - Expectation setting
  - Implementation hints
  - High level design/architecture discussion
  - Ownership of estimate
Security Risk Estimation: Protection Poker

What is the security risk?

Software Security Risk Assessment via Protection Poker

<table>
<thead>
<tr>
<th>Value</th>
<th>Ease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Impact</td>
<td>Lowest Priority</td>
</tr>
<tr>
<td>High Impact</td>
<td>Highest Priority</td>
</tr>
<tr>
<td>Difficult to Exploit</td>
<td>Easy to Exploit</td>
</tr>
</tbody>
</table>

Computing Security Risk Exposure

<table>
<thead>
<tr>
<th>Traditional Risk Exposure</th>
<th>probability of occurrence</th>
<th>X</th>
<th>impact of loss</th>
</tr>
</thead>
</table>

Ease points | Value points

Protection Poker Overview

“Diversity of ideas is healthy, and it lends a creativity and drive to the security field that we must take advantage of.”

-- Gary McGraw

- Calibrate value of “assets”
- Calibrate ease of attack for requirements
- Compute security risk (value, ease) of each requirement
- Security risk ranking and discussion
Diversity of devious, attacker thinking is essential!

Informal discussions of:
• Threat models
• Misuse cases

Memory Jogger

Value Points

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
</table>

Low value

Consider the value of the "asset" with respect to:

Valuable to whom?
- The Company running the software
  - How critical is the process (e.g., how does it contribute to the bottom line)?
  - How long would it take to restore?
  - How harmful is the loss?
- The Attacker:
  - What would benefit from the loss?
  - What can be done with the loss?
  - How much can damage can be done?

Ease Points

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
</table>

Hard to Attack

Consider the following as some criteria for the candidates for hardest to attack:

• Story does not create new pages or user input fields.
• Story reduces the current number of pages or user input fields.
• Exception processing helps to prevent information leakage.

Consider the following as some criteria for the candidates to easiest to attack:

• Story adds new pages.
• Story adds new user input fields.
• Story has few (or none) role(s) with significant read, write, update authority.
• Story requires a significant change in access control (permissions).
• Story provides default usernames and passwords when the product is shipped.
• Story does not enforce strong passwords.
• Story does not have any logging or logging does not identify the specific user.
Security Risk Assessment

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Ease Points</th>
<th>Value Points</th>
<th>Security Risk</th>
<th>Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req 1</td>
<td>1</td>
<td>100</td>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>Req 2</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Req 3</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Req 4</td>
<td>20</td>
<td>5</td>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>Req 5</td>
<td>13</td>
<td>13</td>
<td>169</td>
<td>2</td>
</tr>
<tr>
<td>Req 6</td>
<td>1</td>
<td>40</td>
<td>40</td>
<td>5</td>
</tr>
<tr>
<td>Req 7</td>
<td>40</td>
<td>60</td>
<td>2400</td>
<td>1</td>
</tr>
</tbody>
</table>

Sum of asset value (e.g. one 20 and one 40)

Academic Trial

• 50 students in undergraduate software engineering course

1. Security cannot be obtained through obscurity alone.
2. Never trust your input.
3. Know your system.
4. Know common exploits.
5. Know how to test for vulnerabilities.
Industrial Trial

- Active participation by all on-site team members
- Requirements revised for added security fortification
- Cross site scripting vulnerability found on the spot
- Expressed need for education on cross site scripting
- Expressed need for governance to prioritize security fortification
- Increase awareness of necessary security testing

Protection Poker focuses discussion on what you feel are the true security risk issues

![Bar chart showing percentage of respondents discussing key issues](chart.png)
Rate your software security knowledge

<table>
<thead>
<tr>
<th>1-low</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5-high</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>40</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>

% respondents

Post Tutorial  After two sessions

Protection Poker will help spread security knowledge throughout your team

<table>
<thead>
<tr>
<th>1-not likely</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5-great potential</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>35</td>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>

% respondents

Post Tutorial  After two sessions
Protection Poker will help you learn about software security

<table>
<thead>
<tr>
<th>% respondents</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Post tutorial</td>
<td>5</td>
<td>15</td>
<td>35</td>
<td>45</td>
<td>30</td>
</tr>
<tr>
<td>After two sessions</td>
<td>5</td>
<td>15</td>
<td>25</td>
<td>35</td>
<td>40</td>
</tr>
</tbody>
</table>

(Subjective) Results of Protection Poker

- Explicit result (<20)%:
  - Relative security risk assessment

- Side effects/implicit results (80%+):
  - Greater awareness understanding of security implications of requirement
    - Collaborative threat modeling
    - Collaborative misuse case development
  - Requirements changed to reduce risk
  - Allocation of time to build security into new functionality “delivered” at end of iteration (appropriate to relative risk)
  - Knowledge sharing and transfer of security information