“Soft” Problems for Games Businesses

- Game design
- Storytelling
- Marketing
- Customer experience
The Engineering Approach

- Define your goals
- Come up with an idea of how to meet them
- Perform an experiment to test the idea
- Evaluate the quality of the experiment
- Evaluate the quality of the idea
- Evaluate the quality of your goals
- Repeat
Necessary Ingredients

- The right attitude
- The right people
- Well-defined goals
- Well-communicated goals
- Well devised tests
Setting Goals

- “Product focus” helps you define good goals
- Filter all goals through the lens of customer experience
- Seek helpful constraints
  - Don’t try to solve boundless problems
- The “hardcore gamer” problem
Engineering Game Design

- Goal is a fun game
- Ideas are your game designs
- Playtests are your experiments
- Evaluate your designs as a result of playtests
What does “playtest” mean?

- QA?
- Balancing?
- Focus testing?
- Fun?
Running a Good Playtest

- Are playtesters having the experience you designed?
- Is the experience you designed desirable?
- Learn about things that affect customer experience
  - Game code/NPC behavior
  - Effects art
  - Environmental art
  - Sound
  - Training
  - Pacing
  - Difficulty
Running a Good Playtest

- Make sure the people responsible for the design and execution are there
  - Simplifies evaluation
  - Prioritizes
  - Motivates
- Don’t gather stats!
- Use external playtesters
- Don’t say anything to playtesters
- Ask playtesters to speak through what they are thinking while they play
Questioning Playtesters

- Don’t rely too much on questions
- Oftentimes you learn more from what playtesters don’t experience
- Ask non-leading questions
- Can be great for measuring effectiveness of certain elements
  - Storytelling
  - Perception
Oftentimes this occurs late in production

- Some of your designs work, others don’t
- Fix the most egregious problems

The “legendary” designer

- A designer whose designs always work
- We have no such designers at Valve
Playtesting as Production

➤ Use playtest results to drive production!
  • Create 15 minutes of gameplay in rough form
  • Playtest
  • Use playtest to prioritize work for next week
  • Repeat until complete
Playtesting as Production (Half-Life 2)
Small Increments

- Do the smallest amount that lets you learn something about the player experience
- Use 1-2 week increments
  - Less results in not enough time to make changes
  - More results in churn and flail
- Build about a few hours of game, then start again
  - We felt done as soon as playtesting was no longer painful to watch
Tech development

- **Options**
  - Build a new engine
  - Build off your previous title’s engine
  - Use a licensed engine

- **You do need to do some up-front work**
  - But not too much -- this was our big mistake on Half-Life 2

- **Use iterative tech development**
  - Identify key technology bets – do those first
In Theory

➤ Don’t let theoretical problems prevent playtesting
  • They might not actually be problems
  • If they are problems, the playtest will prioritize which to solve first
  • Playtest may generate ideas of how to solve actual problems better

➤ Don’t discard game designs on theoretical problems
Other Benefits

- Useful for idea generation and learning
- Easy to measure an element’s incremental value or damage
- A great way to avoid design argument
- Can use playtest results to drive other aspects of production
Playtesting as Production

- Solutions to playtest problems can be iterative
- Solve your problems in the right order
- Don’t overcorrect
- Don’t oscillate
- Finish successful elements before moving on
Product-level Benefits

- Allows you to schedule to a particular quality metric
- Scopes game design risk for key features
- Allows you to optimize toward your most successful elements
- Allows you to measure risk, speed, cost
Game Design Experiments

Use them in your games!
What’s most important to our customers?
Is our marketing effective at reaching our customers?
What are the worst problems plaguing our customers?
What perf and memory budgets do we need to meet?
Steam
Building online games

Building features for the customer
• Auto updates
• Anti-cheat
• Communications – e.g., Friends

Building features for your business
• Product encryption/anti-piracy
• Direct sales
• Measurement
Target Platform Experiments

- www.steampowered.com/status/survey.html

<table>
<thead>
<tr>
<th>Video Card Description</th>
<th>Count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATI Radeon 9600 Series</td>
<td>79,818</td>
<td>9.55 %</td>
</tr>
<tr>
<td>ATI Radeon 9800 Series</td>
<td>78,006</td>
<td>9.33 %</td>
</tr>
<tr>
<td>NVidia GeForce FX 5200 Series</td>
<td>66,756</td>
<td>7.98 %</td>
</tr>
<tr>
<td>NVidia GeForce4 MX Series</td>
<td>53,963</td>
<td>6.45 %</td>
</tr>
<tr>
<td>NVIDIA GeForce 6600</td>
<td>45,801</td>
<td>5.48 %</td>
</tr>
<tr>
<td>ATI Radeon 9200 Series</td>
<td>42,291</td>
<td>5.06 %</td>
</tr>
<tr>
<td>NVidia GeForce4 Series</td>
<td>40,443</td>
<td>4.84 %</td>
</tr>
<tr>
<td>NVidia GeForce2 MX Series</td>
<td>27,297</td>
<td>3.26 %</td>
</tr>
<tr>
<td>NVidia GeForce 6800 GT</td>
<td>22,743</td>
<td>2.72 %</td>
</tr>
<tr>
<td>NVidia GeForce FX 5600 Series</td>
<td>20,511</td>
<td>2.45 %</td>
</tr>
</tbody>
</table>
Product Success Experiments

Number of Players vs Time

© 2007 Valve Corporation. All Rights Reserved
Other Experiments

- Number of Steam sales/pre-orders
- Number of registrations (online and retail)
- Number of crashes
Conclusions

- Use the experimental approach today!
- Use playtesting to drive game production
- Steam is one tool that can help