



The Hard Choices Game Explained



The Hard Choices game is a simulation of the software development cycle meant to communicate the concepts of uncertainty, risk, options, and technical debt. In the quest to become market leader, players race to release a quality product to the marketplace. By the end of the game, everyone has experienced the implications of investing effort to gain an advantage or of paying a price to take shortcuts, as they employ design strategies in the face of uncertainty.¹

Nanette Brown

Philippe Kruchten

Erin Lim

Robert Nord

Ipek Ozkaya

Version 1.0
May 19, 2010

INTRODUCTION

While agile methods are very appealing to practitioners, and getting a lot of attention in industry, software development organizations are facing difficulties in applying these methods in projects of increasing scale. A key issue we have identified in large industrial settings is the lack of attention to architectural design and development. One of the tenets of agile software methods is to focus on delivering working software early and often, to demonstrate observable benefits to the end users. This is frequently achieved by focusing on the “skin” of the system, and deferring or ignoring some of the deeper architectural issues. By taking shortcuts, projects incur “technical debt” that grows; some projects may even collapse under the weight if the debt is not recognized and properly managed.

The Hard Choices game was developed to give participants a better understanding of the strategies they employ during software development and the implications of investing effort to gain an advantage or paying a price to take shortcuts. Discussion during the debrief session after the game may touch on the following topics:

- technical debt and investment
- how to assess changing conditions and impediments
- how individual and collective strategies change as projects progress

Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213-2612

Phone: 412-268-5800
Toll-free: 1-888-201-4479

www.sei.cmu.edu

¹ The audience for this report is the facilitator of the game who is using it for educational purposes to communicate principles of technical debt and architectural investment.

The Hard Choices game represents the development cycle for a piece of software. In the quest to become market leader, players race to release a quality product to the marketplace.

PLAYING THE GAME

Hard Choices takes about an hour to play and includes a debriefing session afterward so that players can discuss their playing experiences. The game may be played by two, three or four people. Players compete against each other and race to the finish. The player with the most points at the end of the games wins.

Setting up

The Hard Choices game board represents the activities of a software development release. In their quest to become the market leader, players are competing against each other to release their product to the market place. Players earn points for landing on a square with a tool (representing rewards for investing in technical infrastructure) or by not finishing last (representing rewards for speed to market).

The materials for play include the Hard Choices game board, a six-sided die, markers that players move around the game board, tool cards as counters for rewards, and bridge cards as counters for penalties.

Rules of play

The goal of the game is to accumulate the most points. Players accumulate points by crossing END ahead of their competitors and collecting tool cards. At the end of the game, the person with the most points wins. At the beginning of the game, the facilitator should not inform the players that the game may consist of multiple rounds.

1. One of the positions on the board is marked START and the other is marked END. All players place their markers on the START position. Players begin with zero rewards and penalties.
2. Each player rolls the die to determine who goes first; highest roll wins. Play then proceeds clockwise.
3. During a turn, each player rolls the die to determine how many spaces to move:
 - The player moves his/her piece the number of spaces indicated on the die minus the number of penalties incurred (i.e. the number of bridge cards that the player holds).
 - A player may choose to move in any one direction, including reverse direction, within a single roll of the die. This would increase the opportunity for a player to land on a tool card, at the expense of making progress on the board.
4. The first player to cross END gets 5 points, second gets 3 points, and third gets 1 point. Players also get 1 point for each tool card. To enter the END cell the player should roll anything equal or greater than the number of remaining squares. The game ends when one player remains on the board.

That player stops and doesn't get to take any more turns. He or she gets points for each tool card but doesn't get points for crossing END. The player with the most points at the end of the game wins.

Hard Choices Squares

When a player crosses a "hard choices" square, he or she must decide whether to go over the shortcut bridge or to go the long way and try to collect one or more tool cards.

Hard Choices – Bridges and Tool Cards

1. Bridges count as one movement, similar to squares.
2. A player who chooses to go over a shortcut bridge must collect a bridge card. Each bridge card subtracts one from subsequent rolls of the die.
3. A player may get rid of a bridge card by skipping a turn anytime during the game.
4. If a player lands on a tool square
 - If the player does not already have a tool card, the player gets a tool card.
 - If the player already has a tool card, the player may play a tool card (returning it to the deck) and get a free turn or collect another tool card.
 - A player may only collect one card for a given square within a given round. That is, the player may not use the ability to move backwards to re-cross a tool square and collect a second card for that square.

Do you take the time to gather more tools or do you take a shortcut?

Playing rounds

When the players have finished the first round, the facilitator should announce the second round and add another board (suggesting a second release).

- Plan about 20-30 minutes to play through a round.
- Players play this second round with the rewards and penalties from the previous round in play. This applies to all subsequent rounds.
- To gain understanding, players usually will complete at least two rounds of the game. The facilitator may cut short the second round if he or she feels the participants have understood the underlying concepts of the game.
- The facilitator may announce as many rounds as he or she feels is required for participants to fully experience the underlying concepts of the game. Players may request to play on when they have devised new strategies and wish to see the results. Most likely the game will not run to more than two or three rounds.
- The facilitator may make use of a game-changer prior to the start of each new round (with the exception of the first round). The facilitator can intro-

duce any one of those listed below, or have them printed on cards and have one of the players pick a card at random. Game changers reinforce the role of the rewards (tools) and penalties (bridges) and the uncertainty of conditions. Some suggested game-changers are

- Holders of hammer cards can cross bridges without penalty if they give back the card.
- For each saw card held by a player, add +1 to each die roll.
- For each screwdriver card held by a player, subtract -1 from each die roll.
- Holders of the screwdriver cards give back a bridge card for each screwdriver card owned.
- Each card is now worth higher/lower points; for example, hammer cards are now worth 5 points each.

DEBRIEF SESSION

A debrief session may be conducted at the end of the session, or at the end of each round. Conducting a facilitation session at the end of the first round gives the players an opportunity to discuss analogies to software development and play with that in mind during the second round.

The facilitator can lead discussion by asking

- What just happened? Gather data.
- So what? Generate insight.
- Now what? Figure out what to do differently.

What Just Happened?

Ask players how they experienced the game and what strategies they employed. Some observations might be

- I was learning as I went along.
- If I would have known there would be another round, I would have done things differently.
- I needed to reevaluate my strategy at every roll and take into account where the other players were on the board and how many cards were in play.
- When I saw the other person skipping a turn to retire bridge cards, I did the same thing since I was at a similar spot on the board.
- I took every possible shortcut.
- I kept changing direction during each role to be able to collect as many tools as possible.
- I never turned in a tool card since I thought the points were more valuable than the extra turn.

“Playing Hard Choices stimulated a lot of discussion and reflection on the current real-life strategies for balancing quality, budget / schedule, and features on a project.” – feedback from a game session hosted at Agile Vancouver

Hard Choices is a game to communicate the concepts of uncertainty, risk, options, and technical debt.

- I traveled over the earliest bridge but that didn't help. If I had taken the bridge later I would have been better off.
- The die did not help me and I kept rolling very low numbers, hence no matter what, I could not advance or collect cards.
- Shortcuts hurt more if you take them early, later on they are not so bad.
- Bridges provide short term gain but slow you down in the long term.
- It is human nature to take a shortcut.
- Finishing first doesn't mean you win.

So What?

Ask players to discuss how their experiences in the game relate to the strategies they employ during software development in the face of uncertainty. Ask them how this relates to the choices they make—in investing effort to gain an advantage or paying a price to take shortcuts—and their implications.

Key concepts to discuss include

- Decisions, strategies – ask the “Short-cut Takers” and “Card Collectors” to explain why they choose that strategy. Ask whether they switched categories over the course of the game.
- Release planning – discuss modeling the employment of strategies and the flow of decision making, using the game to simulate software design from the point of view of making (or not making) decisions and the consequences over the course of a software release (or multiple releases). Ask whether the strategies employed are typical of the projects the players are involved in.
- Technical debt, architecture investment – discuss strategies in terms of the categories of technical debt elaborated by Fowler and/or McConnell, in particular short-term versus long-term. Those who sat out a turn to pay back the debt could be employing strategies to maximize their long-term wins, especially in the second round. Ask how architecture can provide options.
- Refactoring, architecture redesign – discuss what this game suggests around strategies for incurring and managing technical debt. Players have the choice to stop and refactor (skip one or more turns depending on the amount of technical debt to be repaid).

Now What?

Ask people what they will take away from the experience and what they might do differently as a result. Some takeaways might include

- A new word to describe a problem or experience, such as technical debt. The metaphor helps people understand the problem and remember the concept.
- Finishing first isn't enough.
- Karma – what you did in the past carries forward.

The metaphor of technical debt helps people understand the problem and remember the concept.

- A better appreciation for the way debt accumulates – not making decisions has consequences and accumulates debt. As one player noted, “I didn’t think that bridge penalties would affect movement the way it did.”
- A new way to communicate with project members and managers about the relationship of investment and debt
- Strategy (anticipation) in the face of uncertainty and competition (adaptation)

Beyond facilitating discussion on the above topics are other options. You might ask participants to write down insights on a post-it and put it on an “Aha wall” at the end of the first round (without discussion), and again at the end of the second round. Then have a discussion. The topic of how to vary the game rules can also result in fruitful discussion with the participants.

VARIATIONS

Discussing variations often comes up during the debrief session. It can be an effective way of extending the analogy and making connections to the software development process. Here are some variations that have been discussed during the debrief sessions.²

Movement

A player may move his/her piece in any direction, but must not switch directions in mid-move. This would decrease the opportunity for a player to land on a tool card.

The game board could be enhanced to include one or two gates, where a player cannot move backwards beyond a gate. A player can move back and forth locally, but there are points beyond which a player cannot recover, requiring them to live with the consequences of early decisions.

Penalties

Different strategies for paying back penalties could be devised. For example, rather than being allowed to pay back a penalty at any turn in the game, players might be allowed only at the beginning of the second round.

² If you do decide to change the game, keep in mind that it works best if you start simple and then layer on additional concepts. We would enjoy hearing from you about your experience.

The Hard Choices game is available for download under a Creative Commons license.

Team Play

Rather than competing individually, players on a game board work as a group and compete against other players on other game boards. The goal is for each team is to get everyone across as quickly as possible with maximum points.

Technical Decisions

Technical decision cards might be used instead of symbolic tool cards for technical participants. Instead of generic tool bonus cards, substitute cards that describe decisions that need to be made at the choice points on the game board. These should be simple, and rely on keywords such as “design a database.” A player draws a card, reads it aloud, and decides to make the decision or not. Making the decision means taking the long way around; not making the decision means taking a shortcut across the bridge.

Stay in Touch

Visit the Hard Choices game community page at:

<http://www.sei.cmu.edu/architecture/tools/hardchoices>

There you can download a copy of the game and share your comments.

Acknowledgments

The Hard Choices game is adapted from the *Short Cut: Game About Speed and Risk* game by Quality Tree Software Inc.

It is developed as part of the Carnegie Mellon Software Engineering Institute Independent Research and Development Project Communicating the Benefits of Architecting with Agile Development in collaboration with the University of British Columbia. For information about the project visit

<http://tiny.cc/saturnarchitectureblog>

We would like to thank Chris Sims for introducing us to creating agile learning games and the short cut game, SuZ Garcia for helping us develop the concept of the game, and our colleagues at the SEI for enthusiastically playing early versions and offering feedback.

Copyright 2010 Carnegie Mellon University.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be directed to permission@sei.cmu.edu.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about SEI reports, please visit the publications section of our website (<http://www.sei.cmu.edu/publications>).