

Game Industry versus Software Industry

Similarities and Differences

Harri Hakonen

harri.hakonen@utu.fi

Department of Information Technology
University of Turku

The 3rd Annual NEXT Conference
Salo, Finland, 5.–6.10.2006

Acknowledgments to:
Jouni Smed and Tuomas Mäkilä

▶ [copyright notice and metapage](#)

Contents

Abstract

There seems to be only few discussions about what is the current [autumn 2006] relationship between game industry and software industry in general. This presentation collects some of the author's experiences and observations from large commercial software development (swd) projects and from academic research on modern swd processes.

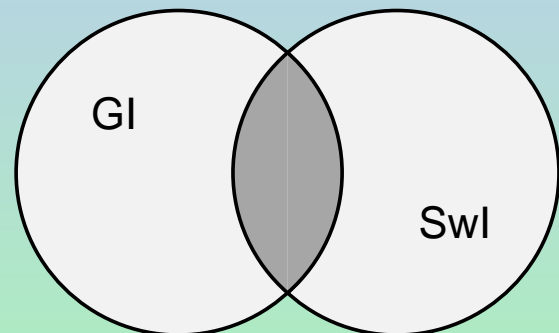
- 1 Introduction: Setting and Approach
- 2 Comparison: Similarities and Differences
- 3 The Future: Challenges
- 4 Our Claim: Migration of Success Ideas



Scope of game industry and software industry

Claim 1

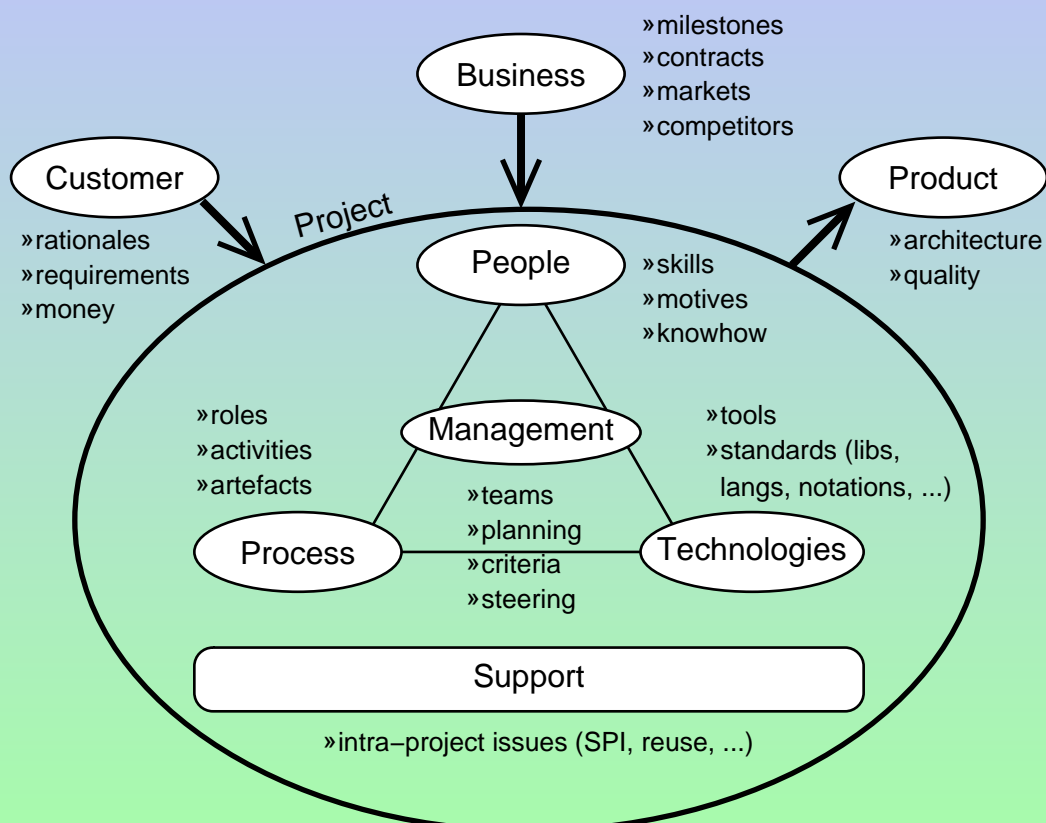
Game industry (GI) is not a subset of typical software industry (SwI): They both have unique aspects.



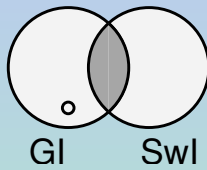
To justify this claim we select a viewpoint: the *attributes* in a *project*.



Typical project attributes



Specific aspects in the game industry



The defining attribute:

The products must have a high “fun” factor.

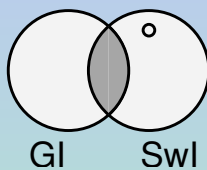
E.g., entertainment and edutainment (i.e. educational entertainment)

Other context specialities:

- product: off-shelf, as-is (extensions, mods, sequels as separate products)
- product lifecycle: short (due to advancing technologies and novelty value)
- markets: publisher centric (‘rock band’ analogy)
- customer: strong community (sharing the experience)



Typical aspects in the software industry



The defining attribute:

The products must have a high utility factor.

I.e., somehow it saves resources or reduces risks

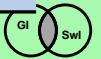
Other context specialities:

- product: customer specific (maintenance agreements)
- product lifecycle: long (need for return of investment, e.g. by stability and baselined infrastructure)
- markets: vendor/customer centric (‘service’)
- customer: values strong integration towards other products (e.g. office applications)



In-project differences in GI and SwI

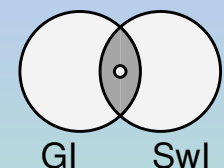
Attribute	GI	SwI
Project size	<ul style="list-style-type: none">• small/few teams on one site, typically ≈ 2 years and 20–50 core persons, narrow variety	<ul style="list-style-type: none">• multiple teams and probably more than one site, wide variety
Sw development process	<ul style="list-style-type: none">• ad hoc, agile-ish, in-house ('film production' analogy)	<ul style="list-style-type: none">• customized from a 'standard' (*UP, Extreme Programming (XP), Crystal, ...)
Technologies	<ul style="list-style-type: none">• state-of-the-art, proprietary	<ul style="list-style-type: none">• mainly standardized, customized
Developers	<ul style="list-style-type: none">• young, basic level of skills is high	<ul style="list-style-type: none">• more aged, skill levels vary



Common aspects in GI and SwI

However, the industries also have similarities:

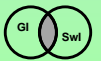
- multiple platforms
 - consoles, PCs, handhelds, embedded hardwares
 - operating systems, third party libraries
- multiuser/multinode applications
 - authentication, billing
 - concurrency management, multiple threads
 - networking issues (consistency vs. responsiveness)
- customizations via parametrization
 - open interfaces (i.e. complete for use but also extendable), e.g. sw engines, service-oriented architecture (SOA)



Common aspects in GI and SwI (cont.)

- hard competition on markets
 - globalization, source of expertise changes
 - expert pools
 - wide variety in customers (both in-product and per product)
- movement towards independent production
 - indie games, open source
 - needs-based applications
- the daily software development activities
 - iterative-incremental practices

It is improbable that a project does not have attributes from $GI \cap SwI$.



Challenges of the game industry

Emerging problems:

- increasing production costs
 - handicraft does not scale linearly
 - demand for more alluring games
- keeping up social fairness
 - cheating, collusion, forgery
 - incomplete legislation (concept of in-game property, digital assets)
- shortage of target-educated labour force — still
 - a team is more than just a group of experts



Challenges of the software industry

Emerging problems:

- over-standardization
 - loss of the competitive advantage (due to encapsulated creativity that makes lateral solutions harder to realize)
- strict *engineering* analogy is problematic in the modern software development
 - the waterfall process assumes a clarity of requirements
 - software 'build' is totally different (in terms of structure, dynamics, and maintenance)
- constant change of tools and work practices
 - the peak of learning curve stays in the horizon
- lots of opinion cliques and narrow-view ideologies (hype waves)
 - there is no 'Holy Grail' or 'silver bullet' for the software development problems¹

¹They all are golden hammers!

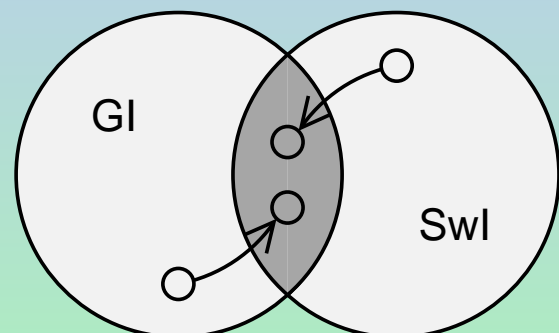


Migration of Success Ideas

Claim 2

There are lessons learned that should belong to both game industry (GI) and typical software industry (SwI).

To justify this claim we give some examples.



Success ideas

I.e. what can be learned from the other side?

The game industry lesson:

It is possible to separate the construction of the system (technicalities) and its content (art).

They

- know how to merge many highly specialized expertise areas (e.g. audio, visual, game design, story) into one
- use problem specific description languages and tools
- have a true portfolio culture (similar to creative arts)



Success ideas (cont.)

I.e. what can be learned from the other side?

The software industry lesson:

It is possible to tackle the software development problems by systematic use of methodologies and standards.

They

- know how to produce appropriate technical architecture from the requirements
- use models for discovering the stable and volatile factors in the given situation
- have a rich set of tried working practices



Summary

We have two claims concerning the relationship between game industry and software industry in general.

First claim is that these industry domains are partly overlapping from the perspective of project attributes. They both have their unique aspects, but also have many commonalities.

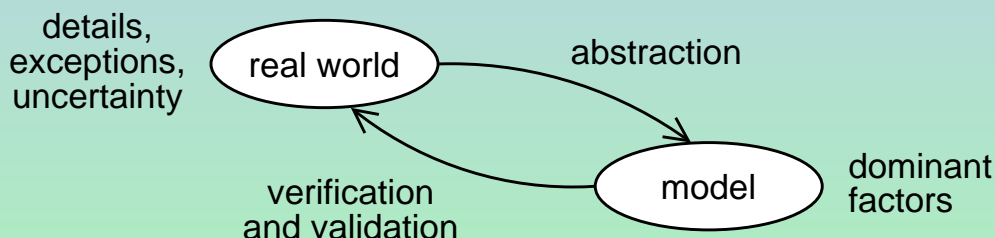
This setting poses the second claim: These industries have lessons learned that can be migrated also to the other area.



Afterwords

What is the role of academic research here?

- catalyst
- synthesizer (big picture and the scale)
- inventor (suggestions)



Socratic method does not know borders!

It is used to find, understand, learn, and refine ideas.



The copyright notice and the metapage of this material:

This material is copyrighted by the authors. All rights reserved. The authors grant permission to copy all or part of this material without fee provided that ● the copies are not sold or traded by any means, and ● at least the authors' copyright, the title of this material, and its date of appearing is given in the form: '©2006 Harri Hakonen: Game Industry versus Software Industry, 2006-10-05'. To copy otherwise, to distribute, or to republish, is not allowed in any form without prior written permission from the authors.

Authors (and Institutes) Harri Hakonen (Dept of IT, Univ of Turku)
Title (and Subtitle) Game Industry versus Software Industry (Similarities and Differences)
Date (year-month-day) 2006-10-05
Version (i.e., compilation date) October 11, 2006
Search Key given by Authors SLIDENEXT320061005a
Updating Strategy Errors are updated only.
Intention This is presentation for The 3rd Annual NEXT Conference in Salo, Finland.

[◀ front page](#)

©2006 Harri Hakonen