

OTS-Based Development

Marco Torchiano

Version 1.1

marco.torchiano@polito.it



© Marco Torchiano, 2007



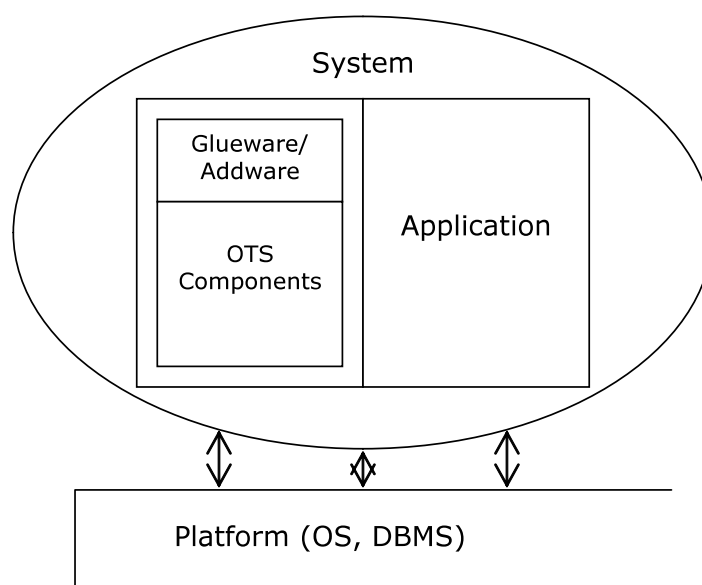
Contents

- Introduction
- Setting the context
- Framework process
 - ◆ Acquisition
 - ◆ Integration
 - ◆ Management
- Empirical results
- Licensing

OTS-Based Development

- An OTS component or product is typically a *domain-specific library (i.e. source or executable code)*
 - ♦ both open and closed source software
- OTS-Based Software is made up mainly of OTS component
- OTS-Based development deals with integrating OTS components into a final product

What is an OTS



Parts of a OTS-based software

- **System:**
 - ♦ Product delivered by the actual project, consisting of application plus components, but excluding platform software.
- **Application:**
 - ♦ Code built in-house to provide certain functionalities of the system.
- **Glueware:**
 - ♦ Code to solve possible mismatch between components, or between components and application or platform.
- **Addware:**
 - ♦ Code to add the functionalities which are expected but not provided by the components.

Relevance of the phenomenon

- Roughly half of all European software projects make use of OTS component
 - ♦ 68% of all software companies use to assemble OTS components
- The use of OSS products is increasing quickly.
 - ♦ EEDC reports that 56% of developers use open source products in 2005, while they were only 38% in 2003.

Why to go the OTS way?

- No effort in development of components
- No skill to develop features
- Ride the technological evolution on the back of the components
- Higher quality of components
- Possibility of switching components
 - ♦ in a plug-and-play dream world

OTS-based systems

- Seminal work done by Carney (1997)
- Types of COTS systems:
 - ♦ Turnkey: only one (suite of) product, with limited customization
 - ♦ Intermediate: one main central product with some satellite products and in-house products.
 - ♦ Integrated: integration of several COTS with equal importance

COTS

- COTS has several *sub-buzzwords*:
 - ♦ **GOTS**: Governmental cOTS
 - ♦ **MOTS**: Modifiable cOTS
 - ♦ **NDI**: NonDevelopmental Item
 - ♦ **OSS**: Open Source Software
- Each subclass presents different issues and problems (Basili and Boehm 2001)

Fact

- Among the papers appeared on journals from 1994 to 2001
- 29% didn't provide any definition of COTS product
- 24% didn't provide any example of COTS products

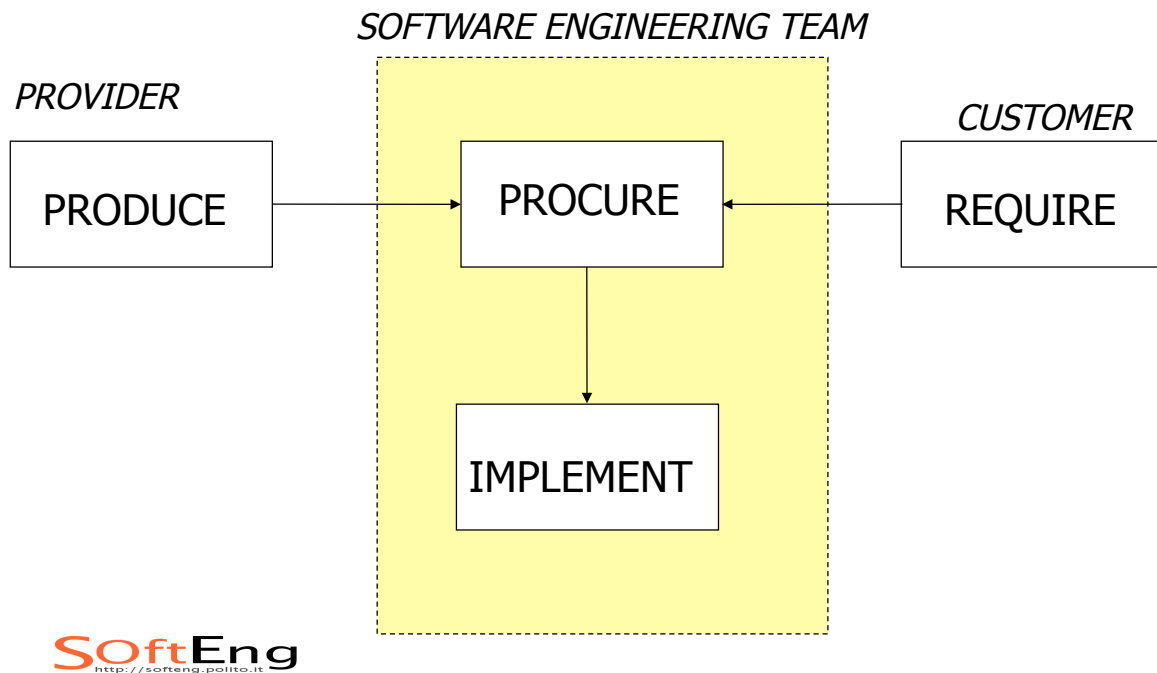
OTS Classification

- COTS: yet another buzzword ?
- Many different meanings:
 - ◆ Windows NT, Oracle, MSOffice, C standard I/O library
- Different problems, risks and methods
- A taxonomy is required to understand what we are talking about

Definition

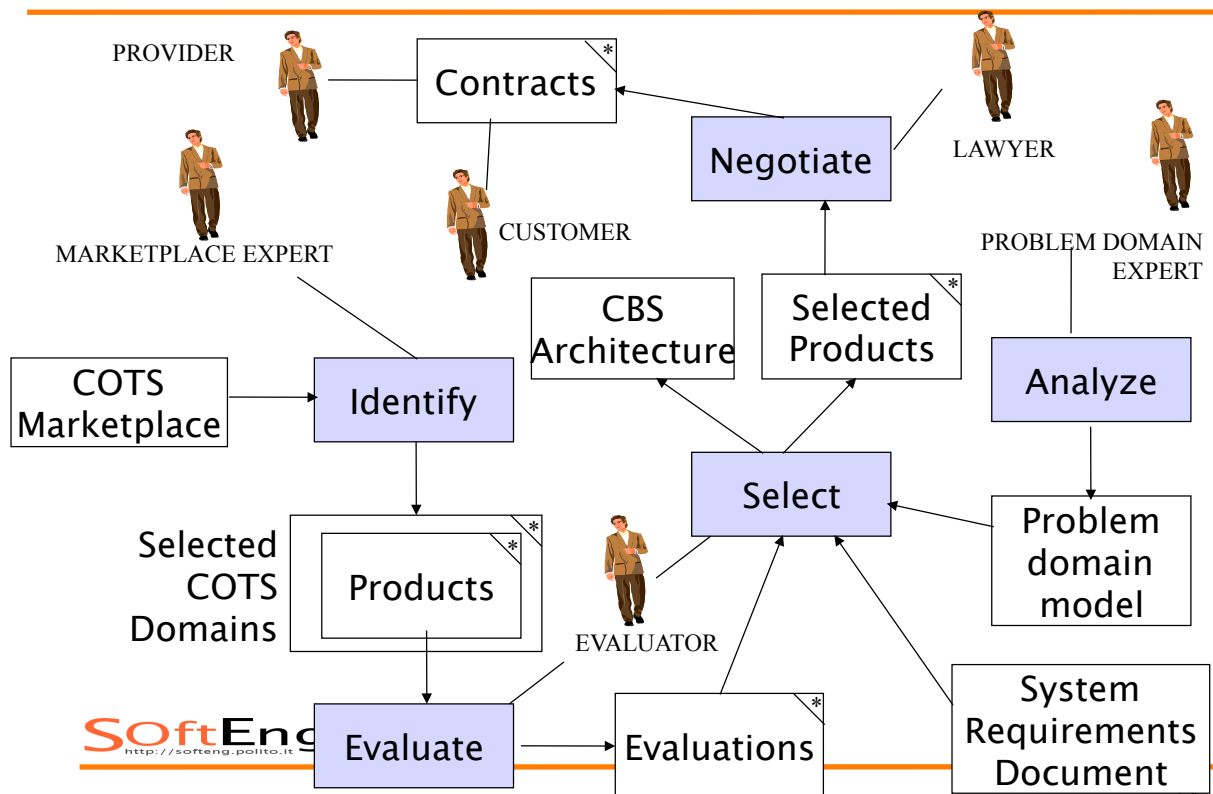
- It is not produced here,
 - ◆ within these projects nor exclusively for them;
- It may be closed-source as well as open source,
 - ◆ open source software usually treated as if it were closed;
- It is not a commodity,
 - ◆ not shipped with the operating system, nor provided with the development environment, nor generally included in any pre-existing platform;
- It is integrated into the final delivered system,
 - ◆ it is not a development tool;
- It is not controllable
 - ◆ in terms of provided features and their evolution.

OBD Context



13

Procurement



14

Procurement

- Identification / Discovery
- Evaluation
- Selection
- Negotiation / Contracting
- The familiarization issue

Selection

- Multi-criteria decision aids
 - ♦ IusWare
 - ♦ OTSO
 - ♦ CAP
- Their purpose is computing a numerical fitness value

WAS

- Essential formula

$$F^Y = \sum_i w_i \cdot e_i^Y$$

- ♦ F^Y : fitness of product Y
- ♦ w_i : weight of feature i
- ♦ e_i^Y : evaluation of feature i on product Y
- Difficult selecting the weights
- Sometimes evaluation is not a number

WAS – Example (1)

- We need to select among three products based on:
 - ♦ Price (-)
 - ♦ Time to fix defects (-)
 - ♦ Time in the market (+)
 - ♦ Response time (-)
 - ♦ Memory requirement (-)

WAS – Example (2)

- For each product and factor the evaluator assign a judgment
 - ♦ Typically a likert ordinal scale is used
 - ♦ Es.{very high, high, medium, low, very low}
- Judgments are transformed into numbers (ratio scale)
 - ♦ very high: 5 ... very low: 1
- Compute fitness of products

Other techniques

- Analytical Hierarchical Process
 - ♦ Pair comparison of criteria and evaluations
 - ♦ Eigenvalues and eigenvectors
- Ranking methods
 - ♦ Outranking of evaluations
 - ♦ Aggregation of evaluations

AHP – Example (1)

- We need to select among three products based on:
 - ♦ Price (-)
 - ♦ Time to fix defects (-)
 - ♦ Time in the market (+)
 - ♦ Response time (-)
 - ♦ Memory requirement (-)

AHP – Example (2)

- Pairwise comparison
 - ♦ What's the relative importance between factors A and B?

comparison	intensity	inverse
equal importance	1	1
light importance	3	1/3
important	5	1/5
very important	7	1/7
absolutely important	9	1/9

AHP – Example (3)

- Comparison matrix with all the possible comparisons (and permutations)
 - ♦ usually matrix is symmetric, but exceptions are possible

	price	time to fix	time in market	response time	memory
price	1	1/3	1/9	1	3
time to fix	5	1	3	1/3	1/5
time in market	9	1/3	1	5	1/3
response time	1	3	1/5	1	5
memory	1/3	5	3	1/5	1

AHP – Example (4)

- We compute the greater eigenvalue λ_{\max} and the corresponding eigenvector u_{\max}
- Then we normalize the vector and obtain the priorities vector (w)
 - ♦ $w = [0.12; 0.18; 0.26; 0.22; 0.22]$

An online eigenvalue / eigenvector calculator:

http://www.arndt-bruenner.de/mathe/scripts/eng_eigenwert.htm

AHP – Example (5)

- Next step is to apply the same procedure to compare the products.
 - ♦ In our example there are five factors:
 - we build five matrices 3 x 3, M_k , each one corresponding to the comparison between the products in relation to each factor.
- We make pairwise comparisons:
 - ♦ Product A has better price than product B?

AHP – Example (6)

- The priorities vector form the columns of a comparison matrix N:

product	price	time to fix	time in market	response time	memory
A	0.55	0.87	0.12	0.07	0.2
B	0.15	0.1	0.38	0.17	0.16
C	0.3	0.03	0.5	0.66	0.64

- The product $N * w$ gives us the ranking:
 - ♦ rank = [0.32; 0.20; 0.45]
 - ♦ i.e. product C is the best

Evaluation attributes

- Can COTS parameters be reused across evaluations and projects?
- There are two contrasting positions
 - ♦ Redefine the measurement framework every time
 - ♦ Define a set of generic attributes that can be reused across projects

No reuse

- Approaches:
 - ♦ Iusware (Morisio et al. 1997)
 - ♦ RCPEP (Lawlis et al. 2001)
 - ♦ OTSO (Kontio 1996)
- Rationale
 - ♦ evaluation is different from simple measurement since it is driven by a well-defined goal,
 - ♦ the factors that influence the choice of attributes depend on the requirements therefore they vary for each project

Reuse

- Approaches
 - ♦ CAP (Ochs et al. 2001)
 - ♦ Boloix (Boloix et al. 1995)
 - ♦ eCOTS (Mielnik et al. 2005)
- Rationale
 - ♦ COTS evaluation is cost effective only if its outcomes can be reused,
 - ♦ using a fixed framework allows evaluating a wider set of products and continuous improvement.

Quality models taxonomies

- Problem
 - ♦ Different types of OTS components require completely different quality models for evaluation
- Solution
 - ♦ Build a taxonomy of OTS components
 - ♦ Taxonomy can be used to select the most appropriate quality model

Quality models + taxonomies

Familiarization

- Fact
 - ♦ Integrators tend to use the products and architectures they already know
- Observation
 - ♦ Familiarity often weights more than objective judgment
- Recommendation
 - ♦ Let the integrators get familiar with several products before selecting

Familiarization

- Playing with products helps drawing them out of the infomog surrounding them and discover their characteristics
- The knowledge of several products favors a more objective judgment
- A pre-selection familiarization reduces the product learning time

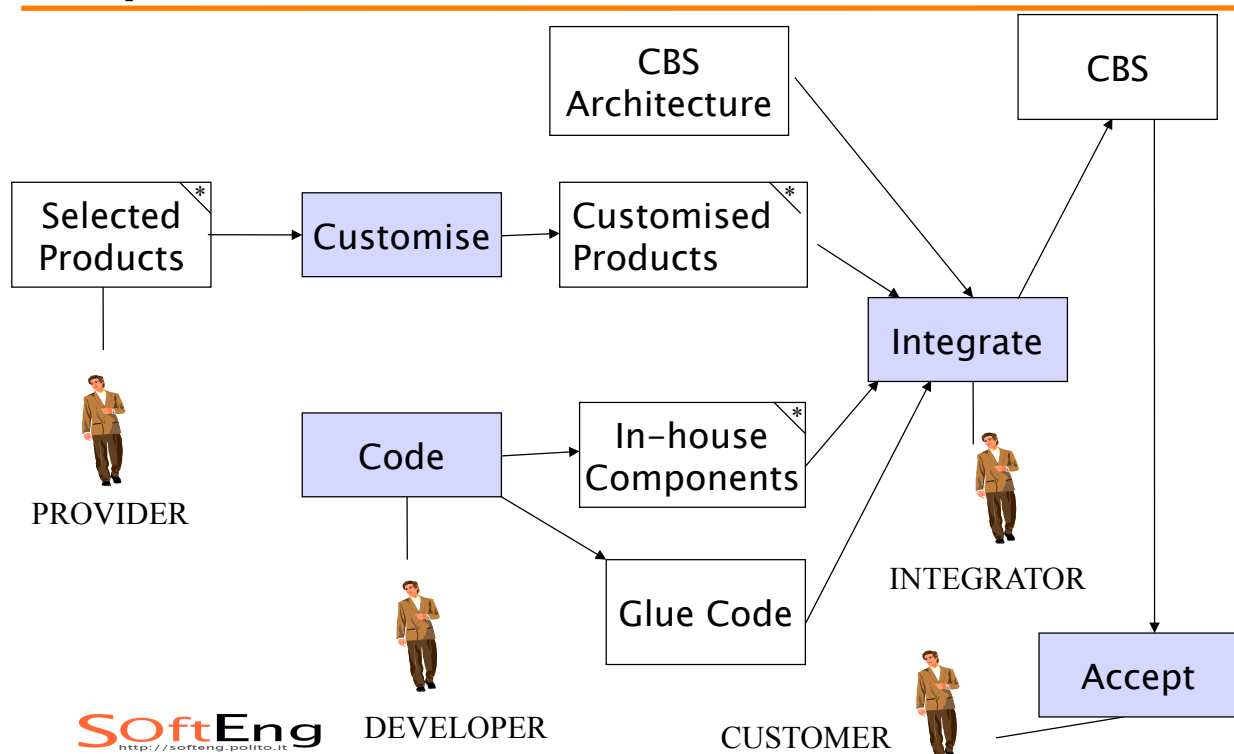
Prototypes and toys

	Consumer	
	Designer	Customer
Familiarize	Toys	
Experiment	Model problems	
Persuade	Model problems	Demo
Elicit		RAD
Develop		Spiral dev.

Playing with toys

- Get it and install it
- Design the simplest application
 - ◆ You need to discover the basic features
- Implement it
- Observe and modify
- Throw away

Implementation



Implementation

- Glue code
- Component technologies
 - ♦ EJB
 - ♦ (D)COM(+)
 - ♦ .NET
- Architectural issues

Architectural mismatches

- Architectural mismatches between components and system
- Components (developers) make implicit assumption about the surrounding context
 - ♦ Packaging (e.g. DLL vs. exe)
 - ♦ Control type (centralized, concurrent)
 - ♦ Information flow (control vs data)
 - ♦ Synchronization
 - ♦ Binding

Architecture

- Fact
 - ♦ Developers often work in the same application domain
 - ♦ The architecture is defined ahead of COTS selection
- Comment
 - ♦ This imposes several limitations
- Recommendation
 - ♦ Draft architecture while selecting COTS

Architecture

- System-level requirements do not always provide hints for the products
 - ♦ Only a (possibly draft) architecture tells whether to use a RDBMS, XML, or text files
- Select an architecture
 - ♦ Reference architectures such as LAMP or .Net make things easier
 - ♦ Sometimes they limit the choice to the complying products

Operation / Management

- Continuous market watch
- Evolution

OTS-based systems evolution

- Upgrades
 - ◆ Products evolve independently from each other (60% of cases)
 - ◆ Hard to switch to new versions at they pop up (2/3 of cases)
 - ◆ There is **no control** over these changes
- Dependencies
 - ◆ Updating a single component (e.g. because of security holes) may cause serious rip effects giving rise to an update runaway

Upgrades (1)

- Integrators tend to delay the upgrade of an OTS component until it is inevitable, usually because
 - ♦ There is some serious bug
 - ♦ The current version is no more supported by the vendor
 - ♦ The customers/users needs are such that new features present in the new version are essential

Upgrades (2)

- There are non-technical solutions to overcome update problems
- OSS:
 - ♦ put your employees at work in the project and try to steer it
- COTS sw:
 - ♦ acquire the vendor or establish a commercial agreement

Dependency Chains

- In the most fortunate cases the dependencies between OTS components are explicitly documented
 - ♦ The chain upgrades are managed automatically.
- Usually the exact dependencies are not always well documented and they form a directed graph (possibly cyclic) that involves a large number of changes.
 - ♦ We have *chain adaptive maintenance*.
 - ♦ Dig into documentation, mailing lists, knowledge bases

Empirical studies

- Characterization attributes as effort predictors
- Structured interviews in Norway + Italy
- Interviews in Norway
- Survey in Europe (Norway, Germany, Italy)

Predicting the integration

- Analysis of two industrial project
- Correlation between characterization attributes and integration process metrics

	Integration Time	Integration Effort
Training time	48%	59%
Functional Coverage	31%	29%

Overlooked facts COTS

- Investigating SMEs in Norway and Italy
 - ♦ Structured interviews
- Two quite different contexts
 - ♦ Pretty good chance to generalize
- Findings often in contrast with common belief
 - ♦ Theses for further research

Open-source

- Belief:
 - ♦ Open-source software is very different from closed source software
- Finding:
 - ♦ Open-source software is often used as closed source

Architecture

- Common Belief:
 - ♦ Architectural mismatches are the main hurdle in composing systems from existing components
- Finding:
 - ♦ Integration problems result from lack of standard compliance; architectural mismatches constitute a secondary issue

Custom code

- Common Belief:
 - ♦ Custom code is mainly devoted to providing additional functionalities
- Finding:
 - ♦ Custom code plays, in the main, an integration role

Selection

- Common Belief:
 - ♦ COTS products are selected (or should be) through a formal selection procedure
- Finding:
 - ♦ Formal selection procedures are seldom used. Familiarity either with the product or with the generic architecture is the key factor in selection

Requirements / Architecture

- Common Belief:
 - ♦ Requirements drive all renegotiation processes involving COTS products
- Finding:
 - ♦ Architecture is more important than requirements for product selection

Control

- Common Belief:
 - ♦ Lack of control over the vendor must be accepted passively
- Finding:
 - ♦ There is a tendency to increase the level of control over the vendor whenever possible

Barriest to OTS-based dev.

- 23 Norwegian industrial persons
 - ♦ Ca. 40% from large-sized IT companies (>100 employees)
 - ♦ Ca. 40% from medium-sized companies (20–99 employees)
 - ♦ Ca. 20% from small-sized companies (<20 employees)
- Industrial concerns and comments have been summarized into five

barriers
 SoftEng
<http://softeng.polito.it>

55

The first barrier

- The barrier to disseminating the overall formal OTS-based development process
 - ♦ Industry is not using OTS-specific processes
 - ♦ It is either too expensive or too difficult
 - ♦ There is no empirical evidence

 SoftEng
<http://softeng.polito.it>

56

The second barrier

- **The barrier to disseminating the formal OTS selection process**
 - ♦ OTS component are not the key parts of the system in some cases.
 - ♦ The formal selection processes are not cost-effective
 - Need to just select a product as fast as possible.

The third barrier

- **The barrier to sharing OTS based development knowledge inside company**
 - ♦ The presence of an OTS knowledge keeper improves projects
 - ♦ There is no clear understanding of how share knowledge about OTS

The forth barrier

- **The barrier to using OSS components**
 - ♦ When using OSS serious problems arise in getting the support reputation information on component providers
 - ♦ Available information of OSS component support come from the previous experience
 - ♦ It is not possible to ensure the future response time

The fifth barrier

- **The barrier to doing requirements (re) negotiation in OTS-based development**
 - ♦ Customer had seldom been involved in COTS component decision making
 - ♦ Because they either had no technical background to make decisions or cared only about the final product

References

- K. Wallnau, S. Hissam. R. Seacord. Building Systems from Commercial Components. Addison–Wesley, 2002.
- M. Torchiano, M. Morisio. "Overlooked Facts on COTS–based Development" In IEEE Software 21(2), March/April , 2004 pp. 88–93.
- International Conference on COTS Based Software Systems (ICCBSS) <http://www.iccbss.org/>

References

1. Jingyue Li, Reidar Conradi, Odd Petter N. Slyngstad, Christian Bunse, Umair Khan, Marco Torchiano and Maurizio Morisio. "Validation of New Theses on Off–The–Shelf Component Based Development" Proc. IEEE 11th International Software Metrics Symposium (METRICS 2004), Como, Italy, September 19–22, 2005
2. Jingyue Li, Reidar Conradi, Odd Petter N. Slyngstad, Christian Bunse, Marco Torchiano, and Maurizio Morisio. "Preliminary Results of a State–of–the–Practice Survey on Motivations of Using Off–The–Shelf Components" In Proc. 6th International Conference on Product Focused Software Process Improvement (PROFES'2005) Oulu, Finland, June 13–16, 2005 pp. 54–68
3. Jingyue Li, Reidar Conradi, Odd Petter N. Slyngstad, Marco Torchiano, Maurizio Morisio, Christian Bunse. "A State–of–Practice Survey on Risk Management in Off–the–Shelf Component–Based Development" In Proc. 4th International Conference on COTS Based Software Systems (ICCBSS), Bilbao (Spain), February 7–11, 2005 pp. 278–288

References

- Carney, D. Assembling Large Systems from COTS Components: Opportunities, Cautions, and Complexities. SEI Monographs on Use of Commercial Software in Government Systems, Software Engineering Institute, Pittsburgh, USA, June 1997
- Carney, D., Long, F., What Do You Mean by COTS?, IEEE Software, March/April 2000, pp. 83–86
- V.Basili, B.Boehm. "COTS–Based Systems Top 10 List" IEEE Computer, Vol. 34, No. 5, May 2001

References

- D.Yakimovich: "A comprehensive reuse model for COTS software products", Ph.D. Thesis Proposal, Computer Science Department, Univ. Maryland, 2001
- Egyed, A.; Medvidovic, N.; Gacek, C. "Component–based perspective on software mismatch detection" In IEE Proceedings–Software, Volume: 147 Issue: 6 , December 2000
- Alessandro Bianchi, Danilo Caivano, Reidar Conradi, Letizia Jaccheri, Marco Torchiano, Giuseppe Visaggio. "Parameters for COTS products selection, an empirical study" In Empirical Methods and Studies in Software Engineering, LNCS 2765, June , 2003 pp. 233–255

References

- Morisio, M. and Tsoukiàs, A.: IusWare: A methodology for the evaluation and selection of software products. IEE Proceedings–Software 144(3) (1997) 162–174
- Kontio, J.: A Case Study in Applying a Systematic Method for COTS Selection. In IEEE–ACM 18th International Conference on Software Engineering (ICSE) (1996) 201–209
- Lawlis, P., Mark, K., Thomas, D. and Courtheyn, T.: A Formal Process for Evaluating COTS Software Products. IEEE Computer 34(5) (2001) 58–63

References

- Ochs, M. A., Pfahl, D. and Chrobok–Diening, G.: A Method for Efficient Measurement–based COTS Assessment and Selection – Method Description and Evaluation Results. In IEEE 7th International Software Metrics Symposium (2001) 285–296
- Boloix, G. and Robillard, P.: A Software System Evaluation Framework. IEEE Computer 12(8) (1995) 17–26
- Mielnik, J. and Laurière, S.: Using eCots portal for sharing information about COTS software products and producers on the Internet and in corporate intranets. MPEC (2005)

References

- M.Torchiano, L.Jaccheri. Assessment of Reusable COTS Attributes. In Proc. of 2nd Int. Conference on COTS Based Software Systems (ICCBSS 2003), Ottawa, Canada, February 10–12, 2003, pp. 219–228.
- Juan P. Carvallo, Xavier Franch, Carme Quer, Marco Torchiano. Characterization of a Taxonomy for Business Applications and the Relationships among them. In Proceedings of Third International Conference on COTS Based Software Systems (ICCBSS), Redondo Beach (CA), February 1–4, 2004, pp. 221–231

Reference

- D. Garlan, R. Allen, and J. Ockerbloom: "Architectural Mismatch or Why it's hard to build systems out of existing parts", Proc. International Conference on Software Engineering (ICSE'95), Seattle, WA, USA, 1995, p. 179–185
- Egyed, A.; Medvidovic, N.; Gacek, C. "Component-based perspective on software mismatch detection" In IEE Proceedings–Software, Volume: 147 Issue: 6, December 2000
- D.Yakimovich, J.Bieman, V.Basili: "Software architecture classification for estimating the cost of COTS integration", Proc. 21st International Conference on Software Engineering, Los Angeles, USA, 1999, p. 296–302.

Licensing Note





Attribution–NonCommercial–NoDerivs 2.5


You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

 **BY:** Attribution. You must attribute the work in the manner specified by the author or licensor.

 **Noncommercial.** You may not use this work for commercial purposes.

 **No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the Legal Code (the full license) found at the end of this document

License (1)

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

- **"Collective Work"** means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.
- **"Derivative Work"** means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.
- **"Licensor"** means the individual or entity that offers the Work under the terms of this License.
- **"Original Author"** means the individual or entity who created the Work.
- **"Work"** means the copyrightable work of authorship offered under the terms of this License.
- **"You"** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- a. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
- b. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats, but otherwise you have no rights to make Derivative Works. All rights not expressly granted by Licensor are hereby reserved, including but not limited to the rights set forth in Sections 4(d) and 4(e).

License (2)

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

- a. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(c), as requested.
- b. You may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation. The exchange of the Work for other copyrighted works by means of digital file-sharing or otherwise shall not be considered to be intended for or directed toward commercial advantage or private monetary compensation, provided there is no payment of any monetary compensation in connection with the exchange of copyrighted works.
- c. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; and to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work. Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.
- d. For the avoidance of doubt, where the Work is a musical composition:
 - i. **Performance Royalties Under Blanket Licenses.** Licensor reserves the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work if that performance is primarily intended for or directed toward commercial advantage or private monetary compensation.
 - ii. **Mechanical Rights and Statutory Royalties.** Licensor reserves the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions), if Your distribution of such cover version is primarily intended for or directed toward commercial advantage or private monetary compensation.
- **Webcasting Rights and Statutory Royalties.** For the avoidance of doubt, where the Work is a sound recording, Licensor reserves the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions), if Your public digital performance is primarily intended for or directed toward commercial advantage or private monetary compensation.

License (3)

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

- a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
- b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

- a. Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
- b. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
- c. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
- d. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.