



G S GRAN SASSO
SCIENCE INSTITUTE

S I CENTER FOR ADVANCED STUDIES
Istituto Nazionale di Fisica Nucleare

Laboratorio di
Ingegneria del Software
a.a. 2013-2014

LEZIONE 11 - Design Antipatterns

Catia Trubiani
Gran Sasso Science Institute (GSSI), L'Aquila
catia.trubiani@gssi.infn.it

Indice

- Contesto: cos'è un design antipattern?
- Antipatterns Reference Model
 - Design Pattern vs AntiPattern concepts
- Catalogo di design antipatterns:
 - Software Development AntiPatterns
 - Software Architecture AntiPatterns
 - Software Project Management AntiPatterns
- Conclusioni e riferimenti bibliografici

Contesto

- Gli antipatterns sono una descrizione in forma testuale di una soluzione che si verifica comunemente per un problema che genera conseguenze negative
- Gli antipatterns sono un metodo per la mappatura in modo efficiente di una "cattiva pratica" per una specifica classe di problemi/soluzioni
- Gli antipatterns forniscono esperienza del mondo reale nel riconoscere i problemi ricorrenti nel software e fornire un rimedio dettagliato per la maggior parte dei problemi più comuni
- Gli antipatterns forniscono un vocabolario comune per identificare i problemi e discutere le soluzioni

3

Origine dei design antipatterns

“The study of AntiPatterns is an important research activity. The presence of ‘good’ patterns in a successful system is not enough; you also must show that those patterns are absent in unsuccessful systems. Likewise, it is useful to show the presence of certain patterns (AntiPatterns) in unsuccessful systems, and their absence in successful systems”

Jim Coplien, 1995

4

Un po' di storia

1979: Brooks, Frederick P., The Mythical Man-Month, Reading, MA: Addison-Wesley

1995: Webster, Bruce F., Pitfalls of Object-Oriented Development, New York: M&T Books

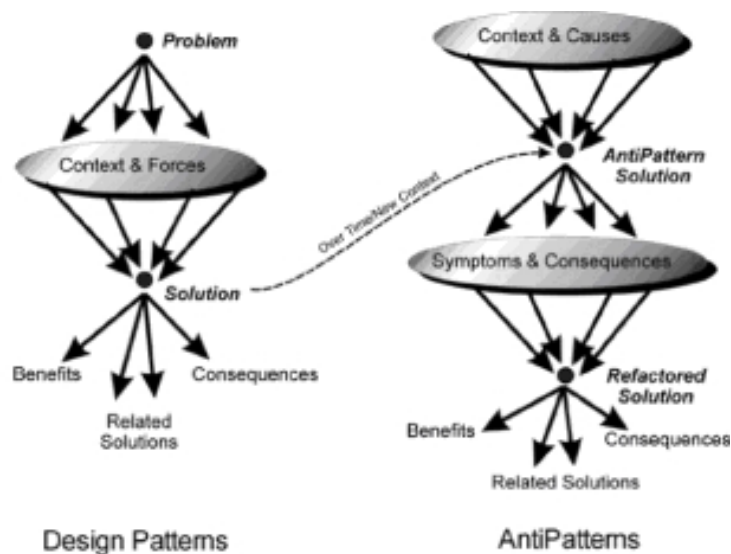
1995: Coplien, James O., "A Development Process Generative Pattern Language", PLoP

1995: Coplien, James O., Object World briefing on design patterns, AT&T Bell Labs Conference Tutorial

1998: W. H. Brown et al. "Antipatterns: Refactoring Software, Architectures, and Projects in Crisis", Addison-Wesley

5

Antipatterns Reference Model



6

Patterns vs Antipatterns

I Design Patterns sono organizzati con il template del pattern per una rapida comprensione della finalità del pattern e della sua soluzione.

I Design Antipatterns sono organizzati intorno al problema, cioè raffigurato come un errore che si verifica comunemente. La soluzione sbagliata massimizza la catastrofe che può essere alla base di un problema fondamentale.

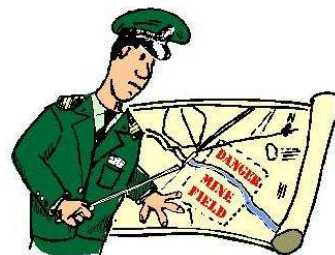
7

Design antipatterns

Antipatterns: Caratteristiche 'negative' di un sistema software

Cosa evitare e come risolvere i problemi

La definizione include le 'bad practices' e come risolverle. L'obiettivo è migliorare problemi comunemente incontrati nello sviluppo del software



8

Tipi di Antipatterns

Software Development AntiPatterns:

- Problemi incontrati dai programmatori
- Riguardano il refactoring come una forma di modifica del codice, usato per migliorare la struttura del software

Software Architecture AntiPatterns:

- Problemi incontrati nella struttura del sistema
- Riguardano problemi comuni ed errori nella creazione, implementazione, e gestione delle architetture

Software Project Management AntiPatterns:

- Problemi incontrati nella comunicazione
- Riguardano progetti software allo scopo di migliorare problemi utente, processi, risorse, e relazioni esterne

9

Software Development AntiPatterns

Un obiettivo chiave di *software development antipatterns* è quello di descrivere le alternative di software refactoring utili. Software refactoring significa modificare codice, si tratta di un'attività che ha lo scopo di migliorare la struttura del software per successive estensioni e/o modifiche a lungo termine. L'obiettivo è quello di trasformare il codice senza comprometterne la correttezza.



10

Software Architecture AntiPatterns

Software architecture antipatterns si concentrano sulla struttura (architettura) del sistema software oppure a livello di applicazioni enterprise e componenti software che ne fanno parte. Nella ricerca scientifica è stato evidenziato che l'esperienza nelle architetture è molto importante nello sviluppo di software.



11

Software Project Management AntiPatterns

Più della metà del lavoro coinvolge la comunicazione umana e risolvere i problemi di comunicazione tra le persone. I *software project management antipatterns* individuano alcuni degli scenari chiave in cui queste questioni sono distruttive per processi di sviluppo del software.



12

Alcuni esempi di Design Antipatterns

Software Development AntiPatterns	Software Architecture AntiPatterns	Software Project Management AntiPatterns
The Blob	Autogenerated Stovepipe	Blowhard Jamboree
Continuous Obsolescence	Stovepipe Enterprise	Analysis Paralysis
Lava Flow	Jumble	Viewgraph Engineering
Ambiguous Viewpoint	Stovepipe System	Death by Planning
Functional Decomposition	Cover Your Assets	Fear of Success
Poltergeists	Vendor Lock-In	Corncob
Spaghetti Code	Wolf Ticket	Intellectual Violence
.....

13

Descrizione del Blob Antipattern

Il Blob si trova in disegni in cui una classe monopolizza l'applicazione, e le altre classi incapsulano dati. Questo antipattern è caratterizzato da un diagramma delle classi composto da una singola classe controller complessa circondato da semplici classi di dati.

Il problema fondamentale è che la maggior parte delle responsabilità sono assegnate ad una singola classe.



14

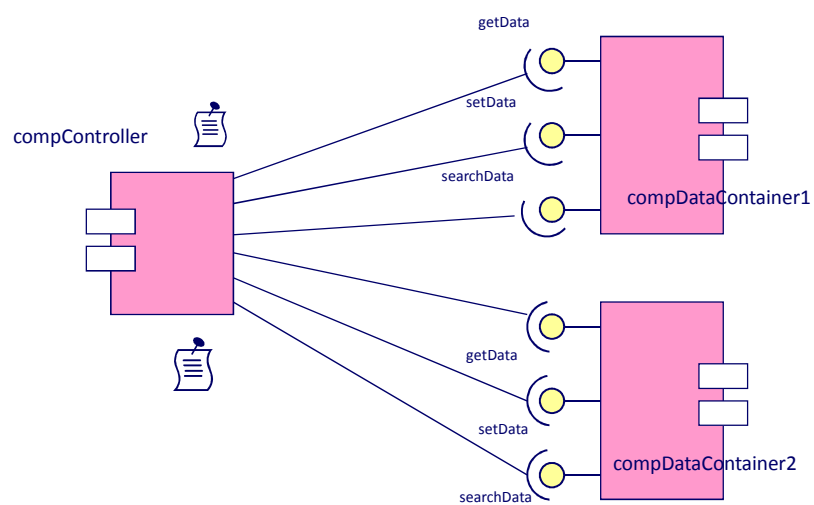
Blob Antipattern

Problema- Design procedurale-style conduce ad un oggetto con numerose responsabilità e la maggior parte degli altri oggetti solo in possesso di dati.

Soluzione- Refactoring del design allo scopo di distribuire le responsabilità più uniformemente e isolare l'effetto delle modifiche.

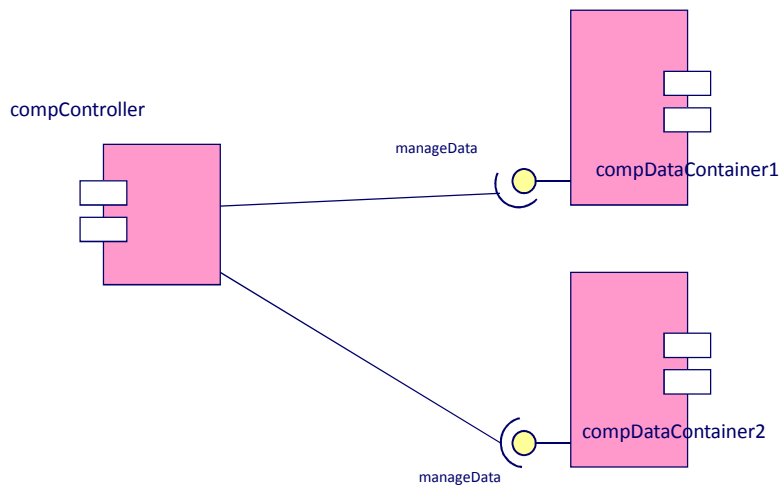
15

Blob: un esempio di problema



16

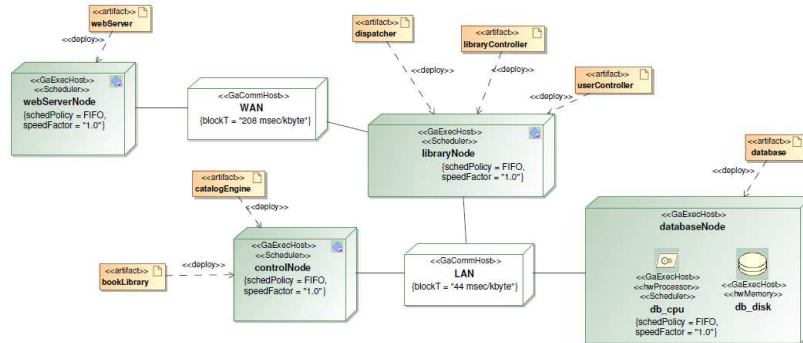
Blob: un esempio di soluzione



17

Come applicare design antipatterns

E-Commerce System (ECS): come individuare le conseguenze negative?



Requirement	Required Value	ECS Observed Value	ECS{blob} Observed Value
RT(browseCatalog)	1.5 sec	1.61 sec	1.44 sec



18

Spaghetti Code antipattern

Spaghetti code antipatterns generano sistemi software con una scarsa struttura. Estensioni e/o modifiche al codice compromettono la struttura del software, anche per lo sviluppatore stesso, soprattutto dopo un certo periodo di tempo.



19

Descrizione di Spaghetti Code antipattern

Se il codice è stato sviluppato utilizzando un linguaggio orientato agli oggetti, il software può includere un numero limitato di oggetti che contengono metodi con implementazioni complesse (molte lines of code - LoC) che richiamano un unico flusso.

Lo scenario tipico è che c'è poca interazione dinamica tra gli oggetti del sistema. Il sistema è molto difficile da 'mantenere' ed 'estendere', e non vi è alcuna possibilità di riutilizzare gli oggetti e i metodi in altri sistemi simili.

20

Conclusioni

Come capire quando si verifica un antipattern? Un antipattern deve:

- Individuare un problema e descrivere come risolverlo
- Ripresentarsi in situazioni rilevanti
- Fornire una comprensione sufficiente per adattare la soluzione
- Avere un nome ed essere riferito in modo coerente

21

Ulteriori letture e riferimenti bibliografici

Altri Design AntiPatterns:

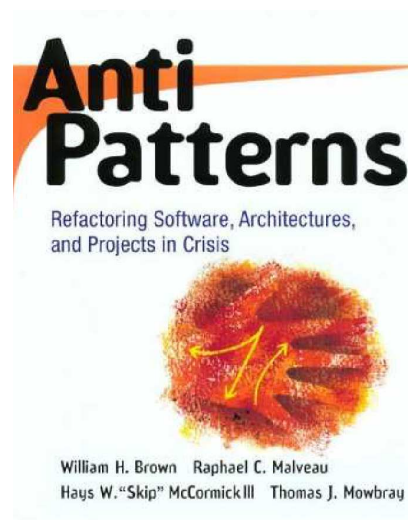
- A Software Development AntiPattern: "Spaghetti code"

Reference principale:

- W. H. Brown et al. "Antipatterns: Refactoring Software, Architectures, and Projects in Crisis", Addison-Wesley, 1998.

Altre references:

- Laplante, P.A. and Neill, C.J., Antipatterns: Identification, Refactoring, and Management. CRC Press, 2005



22

Questions?



catia.trubiani@gssi.infn.it