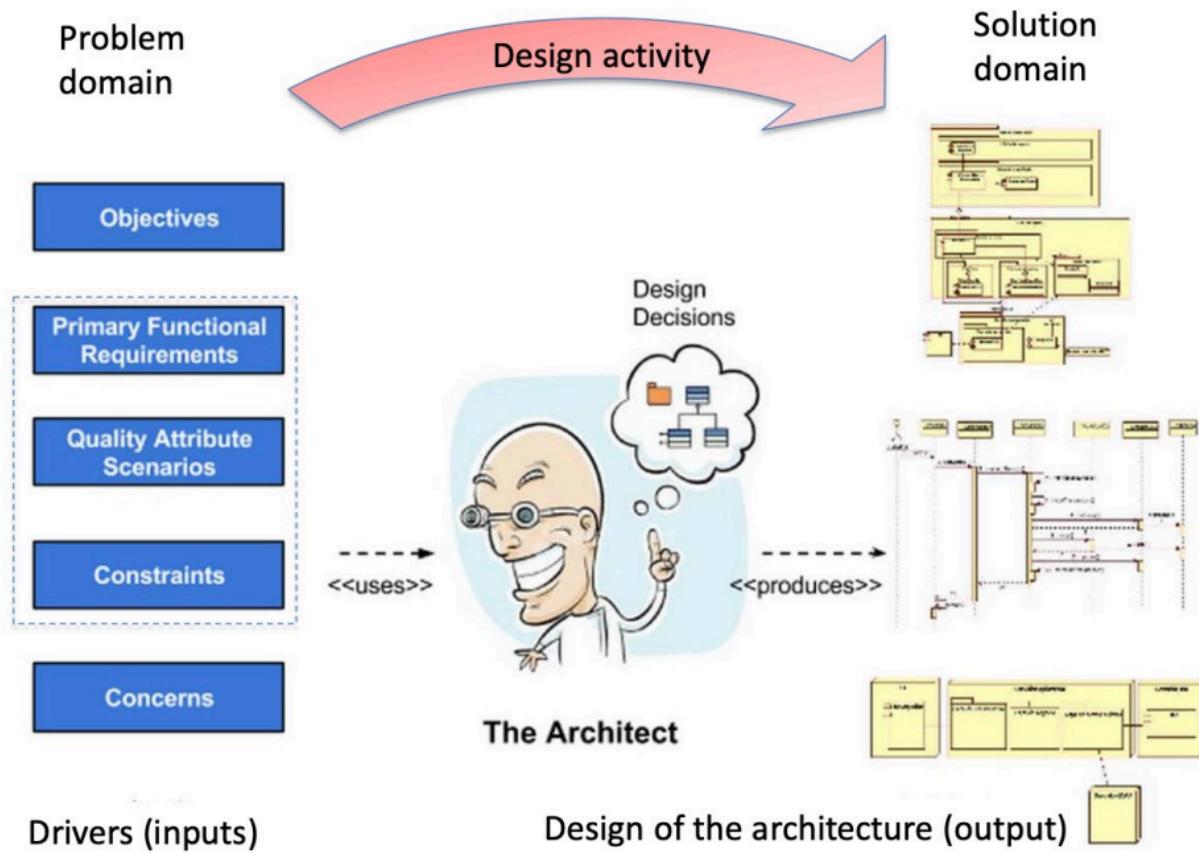


Lezione 11 21/11/2024

Architectural Synthesis - Designing an Architecture

La progettazione architetturale implica prendere decisioni e lavorare con i materiali e le competenze disponibili per soddisfare requisiti e vincoli.



Questi li abbiamo già visti:

- They are a subset of the requirements that *shape* the architecture
 - Functional requirements (Primary Functionality)
 - Quality attribute requirements (Quality Attribute)
 - Constraints (Constraints)

- But other drivers include
 - The type of system that is being designed
 - Design purpose
 - Architectural Concerns



I **sistemi greenfield** sono quelli che devo realizzare from scratch.

I **sistemi brownfield** invece sono quelli dove vogliamo fare delle modifiche ad un sistema già esistente.

Io faccio un design per realizzare un tipo di sistema che avrà un certo tipo di scopo. Faccio un prototipo per fare vedere come funzionerà il sistema (per acquisire il cliente). Oppure devo realizzare un sistema che non si evolverà. Oppure devo fare un sistema che sarà possibile aggiornare, oppure sto proprio facendo un incremento di un sistema.

Io ho dei requisiti funzionali e non funzionali, ma ci sono una serie di altri aspetti che devo tenere in considerazione quando faccio la progettazione del sistema, per esempio:

- Input validation, per esempio in un sistema medico
- Gestione delle eccezioni e registrazione dei log
- Comunicazioni
- Distribuzione e aggiornamenti
- Migrazione e backup dei dati

- Organizzazione del codice
-

Torniamo al metodo da utilizzare per fare il design della soluzione.

Esistono diversi metodi per lo sviluppo dell'architettura:

- Punti di vista e prospettive
- Microsoft
- Processo di Architettura del Software
- ACDM
- RUP
- **ADD (vedremo questo)**

La maggior parte di questi copre l'intero ciclo di vita dell'architettura e offre pochi dettagli su come svolgere l'attività di progettazione.

ADD

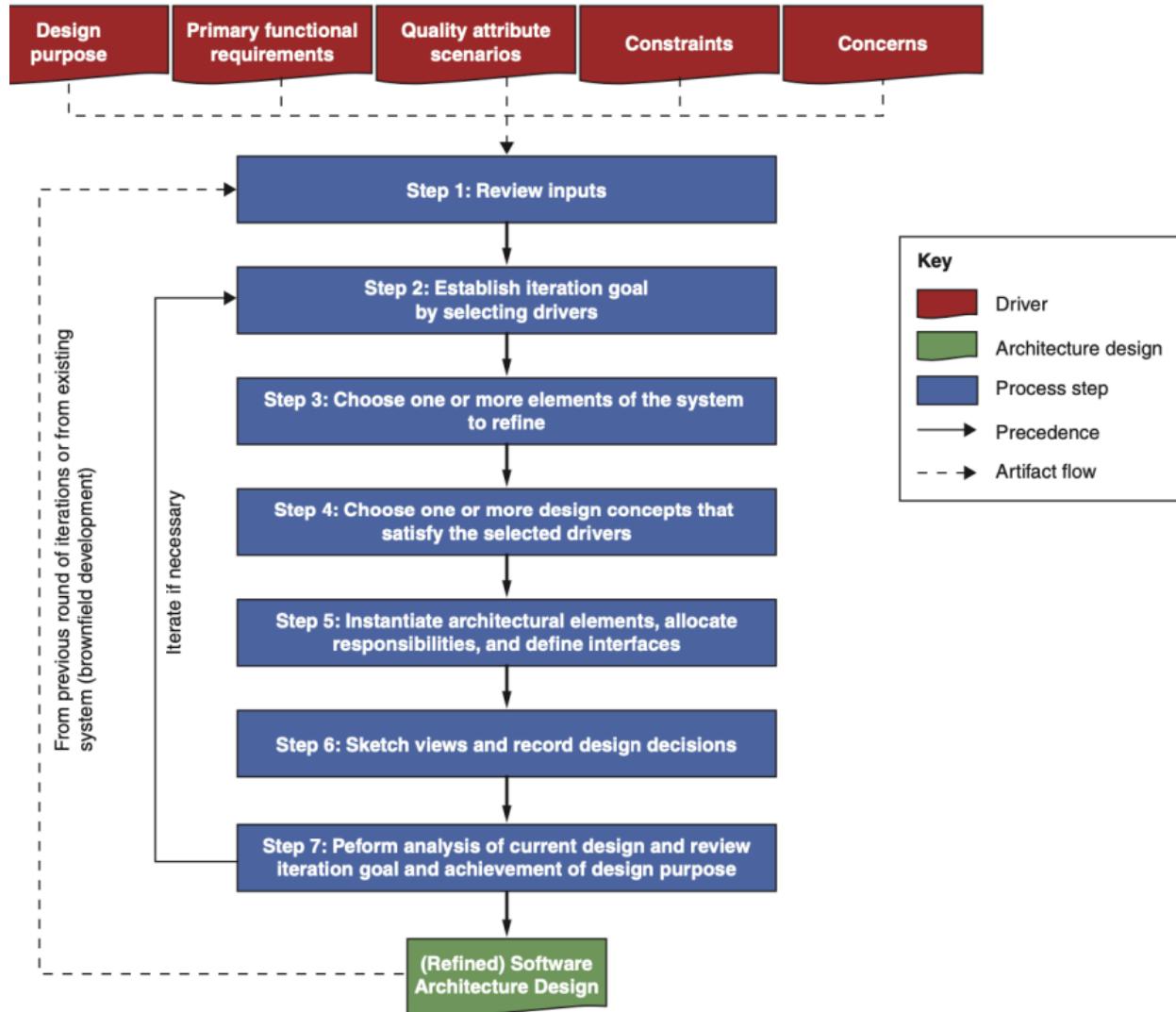
ADD è un metodo di design dell'architettura **attribute driven**, perchè decido qual è l'attributo di qualità che io voglio trattare, quando inizio il processo di definizione dell'architettura.

Questo metodo promuove un **approccio iterativo** al design.

Prevede una serie di step che andiamo a vedere, c'è un concetto di round e un sottoconcetto di interazione.

è un processo strutturato come una serie di passi:

- Identifica gli architectural drivers
- Vado a selezionare la parte di sistema a cui si applicano
- Alloco le funzionalità
- Definisco le interfacce
- Verifico e faccio il refine dell'architettura



Un **round** significa che io vado per step successivi, cioè io voglio soddisfare certi tipi di input nella fase 1, che sono degli ASR, e a questo punto itero finché non li soddisfo. Poi prendo un altro pezzo e faccio un altro round.

Nella prima iterazione si fa la scomposizione in macromoduli, facendo una divisione rispetto alle responsabilità. Poi decido quali sono i requisiti che voglio trattare. Quindi vado a selezionare il componente che li deve gestire e vado a trovare la soluzione architetturale, e iterò finché trovo quella ideale (magari vado a spezzettare in micromoduli etc).

Step 1, review inputs: prima di iniziare un ciclo di progettazione, è necessario assicurarsi che i fattori determinanti dell'architettura siano disponibili e corretti.

Bisogna iniziare a pensare alla soluzione architetturale quando potenzialmente non ci sono ancora tutti i requisiti.

Step 2, stabilire l'obiettivo dell'iterazione: devo selezionare i drivers, ovvero quelli che pilotano la realizzazione di questa iterazione, ovvero dobbiamo scegliere gli scenari specifici che vogliamo realizzare.

Step 3, scegli uno più elementi del sistema da scomporre: per andare a capire quali sono i pezzi che realizzano le funzionalità che ho scelto.

Step 4, scegli un design concept da applicare per risolvere quel driver specifico scelto:

Step 5, devo istanziare gli elementi architetturali, assegnare le responsabilità e definire le interfacce: devo andare a dire quanti layer sono, quali sono le responsabilità di ciascun layer, quali sono le interfacce di comunicazione di ciascun layer. Una volta che ho selezionato uno o più design concept, io devo istanziarlo, ovvero renderlo concreto rispetto al mio requisito che sto cercando di andare ad analizzare.

Step 6, abbozzare le viste e registrare le decisioni di progettazione: le scelte vanno anche motivate. Bisogna registrare ogni decisione particolarmente significativa, registriamo le viste.

Step 7, vado a vedere se la soluzione per affrontare quello specifico driver è stata corretta o meno. Bisogna fare un'analisi per capire se abbiamo affrontato tutto. Dipende a che iterazione e a che round siamo.

Quindi torno su, itero. Prendo un altro scenario, e vado avanti.

Oppure quando faccio la valutazione, anche se ho già trattato tutti gli scenari, mi rendo conto che sono ancora troppo generali, torno su e faccio un raffinamento dell'architettura.

Io itero sia per finire tutti i driver che avevo selezionato, ma anche per andare più in profondità rispetto al disegno che ho fatto di un componente che ho scelto. Però iterare troppo significa perdere tempo prima di poter andare in sviluppo.

Devo cercare di capire se sono abbastanza confidente che quello che ho fatto non ha rischi di fallimento.

Per quanto riguarda il **design decision** si parla di step 3, 4 e 5, serve una conoscenza dei concetti architetturali esistenti ma anche una certa conoscenza

del dominio che stiamo trattando.

Lo step 4, è lo step dove dico ho questo problema, qual è la soluzione architetturale più idonea a realizzarlo?

La maggior parte dei sotto-problemi affrontati durante un'iterazione può essere risolta utilizzando soluzioni esistenti, ovvero concetti di progettazione. È meglio (e più veloce) utilizzare una soluzione comprovata per un problema in cui potremmo non essere esperti.

Come faccio a scegliere il design concept? Devo conoscere le best practices.

A questo punto dovrei scegliere la soluzione architetturale migliore, spesso si utilizzano metodi come lo SWAT (forza, debolezza, opportunità e minacce), sostanzialmente si analizzano i punti di forza e di debolezza.

Sullo step 5: qui dato il concept scelto, vado a concretizzarlo rispetto al mio problema. L'istanziazione significa produrre strutture. Devo capire quali sono gli elementi e quali sono le relazioni tra gli elementi. Poi devo capire come far comunicare tra di loro gli elementi attraverso le interfacce.

Lo step 6 era quello della documentazione, devo raccogliere i diagrammi che ho fatto, devo spiegare perchè li ho fatti così. Devo quindi scrivere come sono arrivato al risultato, le motivazioni della struttura, delle relazioni, delle interfacce.

Allo step 7, alla fine di tutto, occorre fare un'analisi per vedere se quello che ho fatto è sensato. Spesso si usano i backlogs, ovvero delle sorte di todo-list, dove dividiamo quello che va fatto, quello che è in corso d'opera, e quello che abbiamo completato, qui ci va un po' tutto, e aiuta a capire a che livello di definizione dell'architettura siamo.

Valutazione dell'architettura

In questa fase dobbiamo capire se l'architettura descritta ha senso.

Il costo della valutazione deve essere inferiore al valore che la valutazione mi fornisce. Ha senso farlo prima di iniziare a sviluppare, perchè costa meno rispetto a dover rifare.

L'output di una valutazione dell'architettura include l'identificazione delle parti dell'architettura che sono **rischiose**.

Un rischio è un evento che ha un impatto, e ha una probabilità che possa accadere.

Come faccio a capire se esistono delle parti architettoniche che possono essere soggette a fallimento?

Per prima cosa vado a vedere se effettivamente gli ASR sono stati presi in considerazione.

Vado a selezionare quelli di cui sono meno sicuro di aver trovato la soluzione ideale, oppure quelli dove ci sono più soluzioni alternative, e vado a fare questo processo di valutazione. Tipicamente si fa allo step 7, e vado a vedere se i goals che avevo individuato allo step 2 sono stati soddisfatti.

- Every evaluation should include (at least) these steps:
 1. *The reviewers individually ensure that they understand the current state of the architecture*
 2. *The reviewers determine the QA scenarios that guide the review*
 3. *For each scenario, each reviewer should determine whether the scenario is satisfied*
 4. *The reviewers capture potential problems exposed during the prior step*

Chi fa questa valutazione? Ho 3 opzioni:

- L'architetto
- I peers (quelli all'interno del team)
- Gli outsiders (quelli all'esterno del team)

Gli valutatori esterni possono rappresentare un punto di vista oggettivo sull'architettura.

"Esterno" è un concetto relativo e può significare:

- Esterno al progetto di sviluppo
- Esterno all'unità aziendale in cui risiede il progetto, ma all'interno della stessa azienda
- Esterno all'azienda

Gli esterni vengono scelti perché possiedono conoscenze o esperienze specializzate, oppure una lunga esperienza nel valutare con successo le

architetture.

I manager tendono ad essere più inclini ad ascoltare i problemi emersi da un team esterno.

The Architecture Tradeoff Analysis Method (ATAM)

Il metodo **Architecture Tradeoff Analysis Method (ATAM)** viene utilizzato per valutare le architetture software.

L'ATAM è stato impiegato per valutare architetture software in settori che spaziano dall'automotive alla finanza fino alla difesa.

L'ATAM è progettato in modo tale che i valutatori non debbano necessariamente conoscere l'architettura o i suoi obiettivi aziendali.

Questo è un sistema utile quando si chiamano dei valutatori esterni.

- **The evaluation team**

- External to the project whose architecture is being evaluated
- Each member of the team is assigned several specific roles to play during the evaluation
- From 3 to 5 people
- Competent and unbiased outsiders

Role	Responsibilities
Team Leader	Sets up the evaluation; coordinates with the client, making sure the client's needs are met; establishes the evaluation contract; forms the evaluation team; sees that the final report is produced and delivered.
Evaluation Leader	Runs the evaluation; facilitates elicitation of scenarios; administers the scenario prioritization process; facilitates the evaluation of scenarios against the architecture.
Scenario Scribe	Writes scenarios in a sharable, public form during scenario elicitation; captures the agreed-on wording of each scenario, halting discussion until the exact wording is captured.
E-Scribe	Captures the proceedings in electronic form: raw scenarios, issue(s) that motivate each scenario (often lost in the wording of the scenario itself), and the results of each scenario's analysis; also generates a list of adopted scenarios for distribution to all participants.
Questioner	Asks probing quality attribute-based questions.

Poi c'è il gruppo dei **project's decision makers**, sono gli stakeholders che prendono le decisioni come il project manager, l'architetto, il cliente e l'evaluation client.

Gli architecture stakeholders sono gli sviluppatori, i tester, sono quelli che in qualche veste sono interessati all'architettura, perchè da una parte devono creare

il software, lo devono testare, lo devono usare... sono da 12 a 15, a differenza degli altri valutatori loro partecipano solo alla terza fase del processo.

Quali sono gli output di ATAM?

- Un overview generale (complessiva) dell'architettura.
- Un'esplicitazione dei business goals, perchè non è detto che questi siano visti da tutti gli stakeholders.
- Gli ASR saranno prioritizzati e saranno definiti nella formula standard "stimo fino a valore"
- Un insieme di rischi, ovvero i punti critici
- I temi, ovvero le debolezze sistemiche dell'architettura, o del processo, o del team
- Un mapping delle decisioni architettoniche che mappano gli ASR
- Un'insieme di sensitive e trade off points, rispetto alle scelte fatte, come possono impattare sui requisiti

Poi ho anche gli output intangibili:

- un senso di comunità tra gli stakeholders che magari prima non si erano ancora conosciuti
- l'apertura di un canale di comunicazione tra architetto e stakeholders
- Un miglior understanding da tutti i partecipanti dell'architettura, delle sue forze e delle sue debolezze

Fasi di ATAM

ATAM prevede 4 fasi:

Phase	Activity	Participants	Typical duration
0	<ul style="list-style-type: none"> Partnership and preparation: <ul style="list-style-type: none"> - Logistics - Planning - Stakeholder recruitment - Team formation 	<ul style="list-style-type: none"> Evaluation team leadership Key project decision makers 	<ul style="list-style-type: none"> Proceeds informally as required, perhaps over a few weeks
1	<ul style="list-style-type: none"> Evaluation: Steps 1-6 	<ul style="list-style-type: none"> Evaluation team Project decision makers (evaluation client included) 	<ul style="list-style-type: none"> 1-2 days followed by a hiatus of 1-2 weeks
2	<ul style="list-style-type: none"> Evaluation: Steps 7-9 	<ul style="list-style-type: none"> Evaluation team Project decision makers Stakeholders 	<ul style="list-style-type: none"> 2 days
3	<ul style="list-style-type: none"> Follow-up: <ul style="list-style-type: none"> - Report generation and delivery 	<ul style="list-style-type: none"> Evaluation team Stakeholders Evaluation client 	<ul style="list-style-type: none"> 1 week

La fase 0 è preparatoria.

La fase 1 ha 6 step:

- Step 1: il leader del team di evaluators spiega ATAM e come funziona
- Step 2: vengono presentati gli obiettivi di business che hanno guidato la definizione degli ASR. Quindi ho un project decision maker che mostra il sistema dal punto di vista degli obiettivi di business.
- Step 3: l'architetto presenta tutta l'architettura, in modo sintetico, e dice dati i business goals qual è l'architettura sviluppata
- Step 4: l'output di questa fase è la catalogazione delle tattiche e dei patterns che sono stati identificati. Il team di valutazione ha una buona idea di che pattern e tattiche sono state usate dall'architetto, perché hanno studiato la documentazione e hanno ascoltato lo step 3, e quindi sono in grado di creare quest'output di concetti architetturali.
- Step 5: gli utility trees sono usati per identificare gli attributi di qualità che il sistema deve avere per raggiungere i business goals. Questi sono stati menzionati nello step 2, ma qui si può andare nel dettaglio.
- Step 6: a questo punto l'evaluation team può valutare tutto. Vengono esaminati un sottoinsieme degli scenari, quelli più significativi, l'architetto li spiega, e quindi viene valutato se l'approccio utilizzato è sensato, e se c'è nel catalogo

dei pattern estratti nello step 4, e quindi vengono documentate le decisioni architettoniche più rilevanti e catalogati i rischi/sensitivity points.

Al termine del passo 6, il team di valutazione dovrebbe avere un quadro chiaro degli aspetti più importanti dell'intera architettura, delle motivazioni delle decisioni di progettazione chiave e un elenco di rischi, non rischi, punti di sensibilità e punti di compromesso.

A questo punto, la fase 1 è conclusa.

Nella **fase 2** abbiamo anche gli stakeholders.

Lo step 1 viene rifatto.

Poi si inizia dallo step 7, si cerca di andare a catturare quali sono gli ASR rilevanti per questi stakeholders, che non sono mai presenti quando si fa il disegno architettonico.

Si fa una lista prioritizzata di scenari, che poi viene comparata alla lista fatta nello step 5, si va a vedere se sono allineati.

Nello step 8, si rifà tutto il processo, si rifà lo step 6, usando i nuovi scenari creati. Si identificano i rischi.

Lo step 9 è la presentazione dei risultati.

L'ATAM è un processo molto grande, se invece ho un progetto che è più piccolo, allora possiamo usare una versione più leggera.

- For smaller, less risky projects use a Lightweight Architecture Evaluation method based on the ATAM
 - May take place in a **single day**, or even a **half-day meeting**
 - May be carried out entirely by **members internal to the organization**
 - Of course, this lower level of scrutiny and objectivity may not probe the architecture as deeply

Gli step sono ridotti

Step	Hours	Notes
1. Present the method steps	0	<ul style="list-style-type: none"> Participants already familiar with process
2. Review business goals	0.25	<ul style="list-style-type: none"> The participants are expected to understand the system and its business goals and their priorities A brief review ensures that these are fresh in everyone's mind and that there are no surprises
3. Review architecture	0.5	<ul style="list-style-type: none"> All participants are expected to be familiar with the system A brief overview of the architecture is presented via views 1-2 scenarios are traced through these views
4. Review architectural approaches	0.25	<ul style="list-style-type: none"> The architect highlights the architectural approaches used for specific quality attribute concerns This may be done as a portion of step 3
5. Review QA utility tree	0.5-1.5	<ul style="list-style-type: none"> Utility tree should already exist; The team reviews the existing tree and updates it, if needed, with new scenarios, new response goals, or new scenario priorities and risk assessments
6. Brainstorm scenarios	0	<ul style="list-style-type: none"> This step can be omitted as the assembled (internal) stakeholders are expected to contribute scenarios expressing their concerns in step 5
7. Analyze architectural approaches	2-3	<ul style="list-style-type: none"> This step—mapping the highly ranked scenarios onto the architecture—consumes the bulk of the time and can be expanded or contracted as needed
8. Capture results	0.5	<ul style="list-style-type: none"> At the end of an evaluation, the team reviews the existing and newly discovered risks, non-risks, sensitivities, and tradeoffs and discusses whether any new risk themes have arisen

Summary

- If a system is important enough for you to explicitly design its architecture, then that architecture should be evaluated
- The number of evaluations and the extent of each evaluation may vary from project to project
 - A designer should perform an evaluation during the process of making an important decision
 - Lightweight evaluations can be performed several times during a project as a peer review exercise
- The ATAM is a comprehensive method for evaluating software architectures. It works by having project decision makers and stakeholders articulate a precise list of quality attribute requirements (in the form of scenarios) and by illuminating the architectural decisions relevant to carrying out each high-priority scenario. The decisions can then be understood in terms of risks or non-risks to find any trouble spots in the architecture
- Lightweight Architecture Evaluation, based on the ATAM, provides an inexpensive, low-ceremony architecture evaluation that can be carried out in an afternoon