

Software solido e usabile: come integrare ingegneria dell'usabilità e del software

Giorgio Brajnik e Andrea Baruzzo

Dip. di Matematica e Informatica
Università di Udine

e

Interaction Design Solutions srl

www.designcoaching.net

Ordine Ingegneri - Udine – 27 giugno 2014

Scopo del seminario

- ▶ Capire l'**impatto di una buona ingegneria dei requisiti** nel successo di un progetto software
- ▶ Avere una **panoramica** di varie tecniche di ingegneria del software e di usabilità
- ▶ Capire **come innestare** tecniche di usabilità in pratiche di ingegneria del software

Questionario di valutazione

goo.gl/SB1Xvg

Indice

Ingegneria dei requisiti e User-Centered Development (Giorgio)

Features e approcci agili (Andrea)

Design concettuale UI e usabilità (Giorgio)

Domain Driven Design (Andrea)

Model-driven acceptance testing (Giorgio)

Design for Testability (Andrea)

Indice

Ingegneria dei requisiti e User-Centered Development (Giorgio)

Features e approcci agili (Andrea)

Design concettuale UI e usabilità (Giorgio)

Domain Driven Design (Andrea)

Model-driven acceptance testing (Giorgio)

Design for Testability (Andrea)

Ingegneria del software

Scopo

- ▶ sviluppare software
 - ▶ di qualità,
 - ▶ nei tempi e costi previsti,
 - ▶ tale da soddisfare le esigenze dei clienti

Problematiche frequenti

- ▶ **Bassa qualità dei requisiti:** completezza, chiarezza, livello di astrazione, importanza, validità
- ▶ **Bassa qualità del software:** del codice (effetto domino, scarsa flessibilità, basso riuso), dell'interfaccia utente (usabilità, User Experience)
- ▶ **Complessità architetture software:** customizzazioni di una versione, incompatibilità dovuta a legacy, ridondanze tra sistemi
- ▶ **Basso controllo del processo di sviluppo:** qualità del software, sottostime di tempi-budget-personale, non tracciabilità dei requisiti, requisiti non prioritizzati, testing inadeguato
- ▶ **Gestione di outsourcing:** bassa qualità del software, diluizione del know how, testing inadeguato

Cause?

Cause di fallimento (Standish Group 2010):

1. poco input da utenti: 13%
2. requisiti incompleti: 12%
3. requisiti che cambiano: 12%

Nei casi di successo:

1. coinvolgimento utenti: 16%
2. supporto dirigenti: 14%
3. chiarezza nei requisiti: 12%

Quindi

Errori nei requisiti:

1. sono i più costosi
2. sono i più gravi
come impatto sull'applicazione
3. sono tra
le maggiori cause di progetti falliti
4. contribuiscono al ritardo nella
consegna
5. (se non ci sono) contribuiscono al
successo del progetto



Come venirne fuori?

Adottare:

1. un approccio user-centered
2. un approccio di gestione agile
3. principi solidi di progettazione

User Centered Design

1. Coinvolgimento precoce e **comprensione** degli utenti
2. Misure empiriche di usabilità
3. Procedimento iterativo:
 - 3.1 Pianificazione dell'indagine
 - 3.2 Sintesi di un prototipo
 - 3.3 Valutazione del prototipo

Catalogo di tecniche

- ▶ interviste strutturate a soggetti interessati (e utenti)
- ▶ analisi utenti e loro livelli di esperienza; profili
- ▶ task analysis
- ▶ studi sul campo (contextual design)
- ▶ definizioni del problema
- ▶ brainstorming e focus groups
- ▶ personaggi
- ▶ scenari d'uso
- ▶ scrivere casi d'uso essenziali
- ▶ user testing formativo

Caso di studio: CLIMAX

(Per un'agenzia assicurativa)

Benvenuto, [Marco Rossi](#) [Esci] | [Dipendenti](#) | [Amministrazione](#)

[Vendite](#) | [Attività](#) | [Collaborazione](#) | [Tutto](#)
[Home](#) | [Aziende](#) | [Privati](#) | [Iniziative](#) | [Polizze](#) | [Pagamenti](#)

Privati

Nome completo	Telefono Ufficio	Data Inserimento	Utente	Telefono Cellulare	Email
Lucia Agliardi	756867	04-01-2012 09:52am	Marco Rossi	978756653	agliardi.lucia@gmail.com
Marco Borriello	786786	04-01-2012 09:51am	Marco Rossi	45684343	borriello123@yahoo.it
Mario Sconcerti	676456	03-01-2012 05:24pm	Marco Rossi	57867876	sconcerti_m@gmail.com

Compiti

Chiudi	Oggetto	Relativo a	Priorità	Status	Data Inizio	Data Scadenza
<input checked="" type="checkbox"/>	Contattare Agenzia	-	Alta	Non Iniziato	02-01-2012 07:30pm	02-01-2012
<input checked="" type="checkbox"/>	Aggiorna documento XYZ.srl	-	Alta	Non Iniziato	03-01-2012 12:00am	03-01-2012

Aziende

Nome Azienda	Telefono	Numero Fax	Comune di Fatturazione
Falegnameria Bruni			Adegliaccio
Studio dentistico Carli			Cassacco
Lemon Software	978464564		

Riunioni

Chiudi	Oggetto	Relativo a	Data Inizio	Accetti?
<input checked="" type="checkbox"/>	Suddivisione clienti	-	10-01-2012 04:00pm	Accettato con riserva
<input checked="" type="checkbox"/>	Resoconto	-	11-01-2012 05:30pm	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

[Stampa](#) | [Torna all'inizio](#)

© 2004-2011 SugarCRM Inc. The Program is provided AS IS, without warranty. Licensed under [AGPLv3](#).
 This program is free software; you can redistribute it and/or modify it under the terms of the
[GNU Affero General Public License version 3](#) as published by the Free Software Foundation, including the additional permission set forth in the source code header.
 SugarCRM is a trademark of SugarCRM, Inc. All other company and product names may be trademarks of the respective companies with which they are associated.



Esempio di formulazione di problema

2.2 Formulazione del problema	
Il problema	della gestione dei dati di clienti, agenti e collaboratori dell'agenzia, della registrazione e il monitoraggio delle loro attività, e del supporto ad attività tipiche giornaliere
riguarda	tutti i membri dell'agenzia stessa, durante le loro attività sui sistemi informatici, che tipicamente sono quelli di inserimento, gestione, reperimento di informazioni dal database e gestione del rapporto con la clientela.
L'impatto del problema è	che le attività di aggiornamento e utilizzo del database sono poco organizzate, ripetitive e portano a incompletezza ed errori nel lavoro. Le attività interne all'agenzia non seguono un protocollo preciso, i sistemi attuali sono utilizzati parzialmente ed il rapporto con la clientela può risentirne. metterei in luce anche la scarsa flessibilità
Una buona soluzione	consiste in un sistema centralizzato accessibile via web con un browser; il sistema deve permettere di accedere ai dati presenti sul database centrale dalle diverse postazioni dell'agenzia, dalle figure professionali presenti, con determinati permessi o restrizioni. Il sistema deve inoltre fornire supporto alle diverse figure professionali nei loro compiti, in maniera semplice ed intuitiva, senza la richiesta di competenze informatiche specifiche.
comporta i seguenti benefici	non sono richieste particolari capacità informatiche per utilizzare il sistema, i tipici compiti da svolgere nel corso della giornata vengono agevolati e resi quindi più veloci, è più semplice la condivisione di informazioni, l'utilizzazione e la comprensione dei dati, il loro monitoraggio e la gestione in generale. L'accesso al sistema non è legato ad un singolo computer. anche qui: maggior flessibilità ci si aspetta porti a più facile e frequente fare nuove campagne, iniziative; marketing + mirato

Esempi di stakeholder

Nome	Ruolo
Agente	Si occupa della gestione e della interazione con i clienti dell'agenzia, prepara e sottopone nuove offerte e campagne ai clienti. Ha quindi la responsabilità sull'intero portafoglio clienti dell'agenzia. Effettua inoltre il monitoraggio della contabilità dell'agenzia, ed è interessato a conoscere la situazione globale dei movimenti dell'agenzia.
Subagente	Ha a propria disposizione un portafoglio clienti limitato, con il quale interagisce allo stesso livello di un agente. A differenza dell'agente, non ha accesso a tutti i dati dell'agenzia, ma solo una parte di essi.
Segretario	Si occupa dell'aggiornamento dei dati anagrafici del portafoglio clienti dell'agente, prepara i documenti necessari all'agente per interagire con i clienti. Aggiorna la contabilità giornalmente, e carica i dati sul sistema generale dell'azienda. Utilizza spesso il telefono per ricevere ed effettuare chiamate, al fine di mantenere un rapporto più stretto possibile con i clienti
Amministratore	Gestisce il sistema in uso, aggiorna e modifica i permessi per gli utenti singoli o gruppi di utenti, inserisce nuovi campi nelle tabelle dati.
Configuratore	Figura tecnica esperta nella modifica del sistema, si occupa di applicare delle personalizzazioni minori al sistema, come l'inserimento di nuovi campi di dati all'interno delle schede cliente, modifica di etichette associate a dati o di componenti grafiche di layout del sistema.
Tecnico	Figura tecnica altamente esperta e conoscitrice del sistema, sia a livello di front-end che di back-end. E' in grado di risolvere i bug del sistema, di implementare nuovi moduli e di fornire supporto a 360 gradi. monitora il funzionamento del sistema, e periodicamente lo aggiorna.

Esempio di personaggio

3.2.4 Maria, segretaria

Maria è la segretaria di una filiale dell'agenzia, da 15 anni. Fino a qualche anno prima, in agenzia gli strumenti informatici erano utilizzati davvero poco, quasi tutta la documentazione era cartacea. Ora invece, l'agenzia si sta informatizzando sempre più, le applicazioni software sono molte e richiedono diversi tipi di competenze.

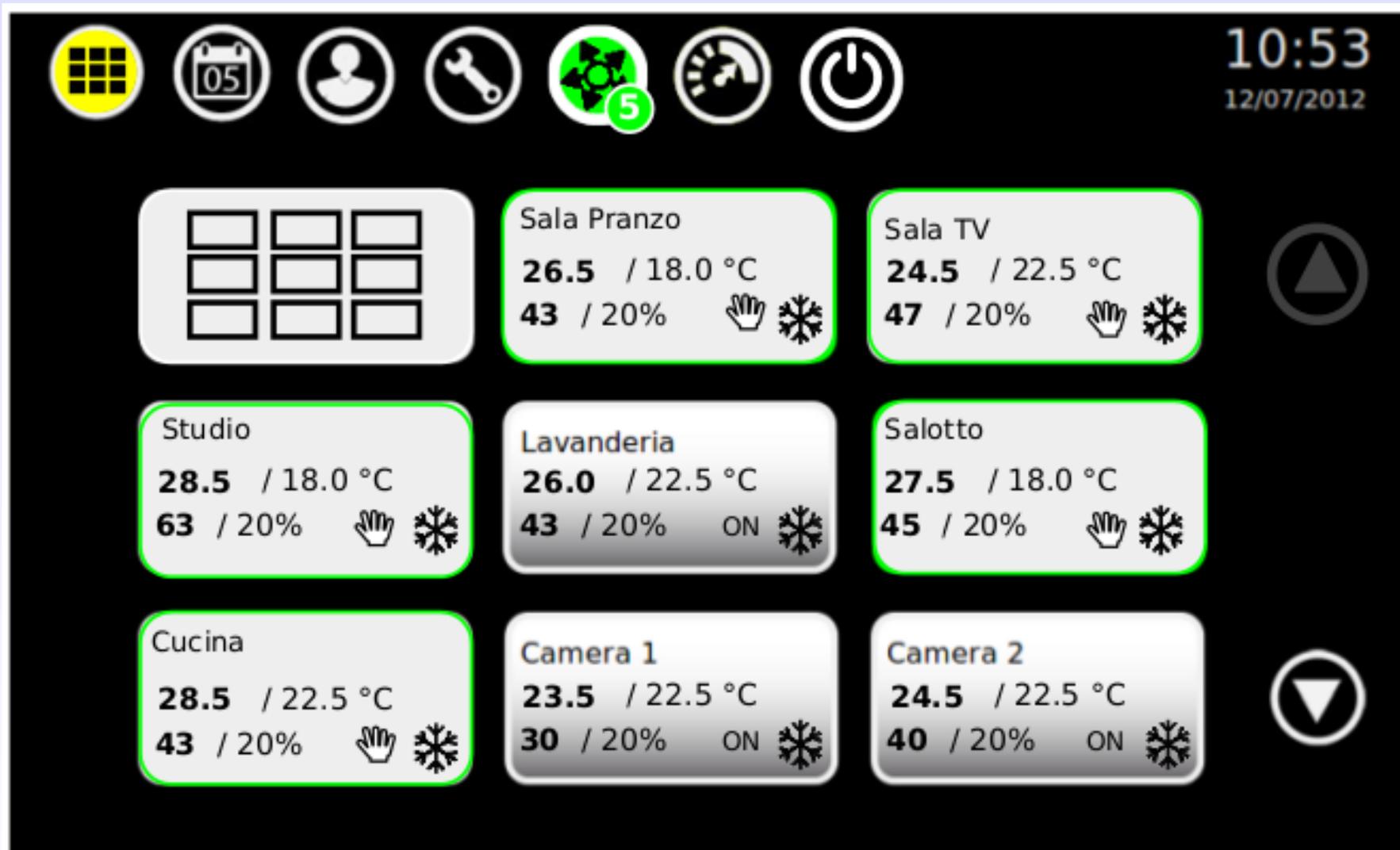


Maria, nelle condizioni attuali, dovrebbe utilizzare durante la giornata diverse applicazioni, saltando continuamente da un'interfaccia all'altra: alcune più conosciute, come gli strumenti di word processing (Microsoft Word), altre, come l'interfaccia da riga di comando per aggiornare il sistema della compagnia, molto più complicate e scomode. Ogni singola operazione richiede molto più tempo di quello che dovrebbe, e così facendo, spesso Maria non riesce a terminare il proprio lavoro giornaliero in tempo, con la conseguenza di dover fare molte ore di straordinario. Ci potrebbe essere anche bassa motivazione, che tanto per lei che faccia o non faccia qualcosa bene e velocemente non influisce sullo stipendio. Che motivazioni potrebbe avere? Promozione?

Maria ha la necessità di utilizzare un sistema che praticamente non richieda alcun addestramento, e che sia facile ed intuitivo nel suo utilizzo. Gradirebbe che in qualche modo si resolvesse il problema della grande quantità di documenti cartacei che ogni giorno deve consultare.

Caso di studio: OTOUCH

(Eurapo Srl e Aragon Engineering Srl)



Esempio di profilo utenti

4.3 Installatore	
RUOLO	<p>installatore</p> <p>che realizza l'impianto e lo configura inizialmente.</p> <p>In pratica si tratta di due figure professionali diverse: l'idraulico, che decide che strumenti mettere in quale stanza, e l'elettrotecnico che stende le connessioni elettriche ed elettroniche. Normalmente inizia l'idraulico con il posizionamento degli strumenti, segue l'elettrotecnico che li collega, e poi di nuovo l'idraulico che imposta la programmazione.</p>
Caratteristiche personali	<ul style="list-style-type: none"> • idraulico/elettrotecnico • conosce bene certi strumenti, altri (nuovi o di fabbricanti poco noti a lui) di meno. Procedo per analogia o per tentativi. • conosce un po' di domotica (soprattutto l'elettrotecnico) • alcuni sono a loro agio con pc, tablet, smartphone; altri no <p>Prevediamo due livelli di esperienza:</p> <ul style="list-style-type: none"> • personale di fabbrica Eurapo (competente/esperto²) • personale esterno (principiante avanzato)
Esperienza specifica in Otouch2	<ul style="list-style-type: none"> • scarsa, principiante avanzato • ha seguito una demo, un mini corso • dato l'uso frequente però potrebbe diventare competente
Interazione	<ul style="list-style-type: none"> • 2-3x al mese, di 1gg, 1x per impianto • uso sporadico, non continuo, a momenti intenso • sessioni di 1-2h • non è guidato da un processo; non esiste una checklist da seguire; mancano anche procedure di collaudo • può operare insieme a supervisore
Ambiente	<ul style="list-style-type: none"> • in sala termica o in altri locali, in piedi, si sposta • forse scarsa luminosità • usa tablet o laptop, no mouse

Esempio di profilo utenti II

Flusso Informazioni	<ul style="list-style-type: none"> • input=progetto impianto, progetto edificio, specs strumenti, caratteristiche di utilizzo dell'impianto (preferenze) Molte di queste informazioni non sono esplicite (ad es. progetto impianto) • output=valori per parametri strumenti • output=nuove programmazioni per impianto (stagioni, scenari) • output=rapporto di installazione, con dettagli sulla configurazione • potrebbe aver bisogno di altre info (di confronto su parametri, cosa fanno che effetto hanno, cosa vogliono dire, come lo si ottimizza)
Goal comportamen- tali	<ul style="list-style-type: none"> • non fare brutta figura con la controparte installatore (idraulico nei confronti di elettrotecnico o viceversa) • non fare brutta figura con supervisore/cliente => fare bene le cose • fornire un impianto che funziona bene, per la parte di sua competenza • essere veloce
Fattori usabilità	<ul style="list-style-type: none"> • velocità nello svolgimento delle attività => non esplora UI, non gradisce feature che non gli servono, reattività • qualità del lavoro prodotto => affidabilità del sistema • uso frequente => è power user, gradisce macro e personalizzazione; gradisce opzioni e settaggi personalizzati (es di parametri per certi tipi di strumenti) • uso saltuario o atteggiamento negativo => facilità di uso, di comprensione, funzionalità limitate, funzionalità guidate • efficacia ==> minimal surprise, commensurate effort, affidabilità

Esempio di caso d'uso essenziale

Essential use case	TCS: Test cablaggio sistematico
Actor	installatore elettricista o idraulico, su richiesta di installazione dell'impianto
Context	da TFT, Desktop o tablet.

	Intention	Responsibility
1	lanciare una procedura automatica di verifica di cablaggio corretto tra scheda termica e tutti gli strumenti	<ul style="list-style-type: none"> individuare situazioni in cui 2 strumenti hanno lo stesso ID individuare (o meglio ancora, prevenire) situazioni in cui gli ID hanno valori fuori range gestire situazioni in cui inversioni di connessioni rx/tx sono presenti fare un ping tra TFT e ciascuno strumento provare ad accendere e spegnere ciascun strumento
2	poterla sospendere, ripristinarla o annullarla	<ul style="list-style-type: none"> consentire di farlo
3	esaminare l'esito	<ul style="list-style-type: none"> generare un rapporto di test in cui si elenca cosa è stato fatto e che esito ha avuto

Indice

Ingegneria dei requisiti e User-Centered Development (Giorgio)

Features e approcci agili (Andrea)

Design concettuale UI e usabilità (Giorgio)

Domain Driven Design (Andrea)

Model-driven acceptance testing (Giorgio)

Design for Testability (Andrea)

Indice

Ingegneria dei requisiti e User-Centered Development (Giorgio)

Features e approcci agili (Andrea)

Design concettuale UI e usabilità (Giorgio)

Domain Driven Design (Andrea)

Model-driven acceptance testing (Giorgio)

Design for Testability (Andrea)

Demo di user testing formativo

Indice

Ingegneria dei requisiti e User-Centered Development (Giorgio)

Features e approcci agili (Andrea)

Design concettuale UI e usabilità (Giorgio)

Domain Driven Design (Andrea)

Model-driven acceptance testing (Giorgio)

Design for Testability (Andrea)

Indice

Ingegneria dei requisiti e User-Centered Development (Giorgio)

Features e approcci agili (Andrea)

Design concettuale UI e usabilità (Giorgio)

Domain Driven Design (Andrea)

Model-driven acceptance testing (Giorgio)

Design for Testability (Andrea)

Test di accettazione

(UAT=User Acceptance Testing)

- ▶ Ha lo scopo di collaudare end-to-end un sistema interattivo
- ▶ Un caso di test descrive
 1. come configurare il sistema sotto test
 2. che input fornirgli
 3. che risposte aspettarsi
- ▶ Una test suite viene sviluppata in base a **rischi di qualità**
- ▶ Spesso i test case vengono eseguiti (sommariamente) a mano
- ▶ Spesso sono informali e poco sistematici (smoke test)
- ▶ Raramente vengono svolti in test di regressione

Model Driven Software Engineering

- ▶ Uso di modelli per **astrarre** aspetti importanti
- ▶ E usarli per produrre **automaticamente** codice eseguibile

Benefici attesi

- ▶ Riutilizzo del modello
- ▶ Modello più facile da capire
- ▶ Manutenzione più semplice usando i modelli

Un approccio per il testing di accettazione

- ▶ Modelli UML per i dati e la dinamica dell'interfaccia utente
- ▶ Generazione automatica di infrastruttura di automazione dei test
- ▶ Semplificazione, robustezza e flessibilità dei casi di test automatizzati

Benefici

1. Velocità nel creare nuove test suite
2. Velocità nell'eseguire nuovi test-run
3. Maggiore qualità esterna dell'applicazione

Caso di studio: www.generatel.it

Calcolo preventivo

The screenshot displays the Genertel website interface for car insurance. At the top, there is a navigation bar with the Genertel logo, a menu with items like 'Assicurazioni', 'Come fare per', 'In caso di incidente', 'Novità', and 'Contatti', and the GenertelLife logo. A left sidebar contains a vertical menu with categories: 'Auto' (selected), 'Moto', 'Furgoni & Van', 'Altri Veicoli', and 'Casa'. The main content area is titled 'ASSICURAZIONE AUTO' and features a red car image. Below the title, it states: 'Assicurazioni Auto online per i veicoli di famiglia ad un prezzo conveniente, con la sicurezza, il servizio e l'assistenza di Genertel!' and includes a link 'Scopri l'assicurazione auto'. A central section titled 'IL TUO PREVENTIVO AUTO IN UN MINUTO' includes a '1'' icon, a sub-headline 'Inserisci i dati richiesti e ottieni velocemente il tuo preventivo', and two input fields: 'Targa Veicolo' and 'Data di nascita del proprietario'. A green button labeled 'FAI UN PREVENTIVO' is positioned below these fields. A note below the button says 'Se non ricordi i dati o preferisci non indicarli [clicca qui](#)'. At the bottom of this section, it says 'Se hai già fatto un preventivo e vuoi recuperarlo [clicca qui](#)'. Below the main content, there is a section 'Premiati per la qualità dei servizi' with 'OTTIMA' award logos and the text 'Nel 2014 Genertel è stata premiata'. On the right side of the page, there are four vertical buttons: 'Pensione' with a red ribbon icon, 'Vita' with a red heart icon, and 'Risparmio' with a red piggy bank icon.

Tecniche di Record and Play

(ad es. con Selenium IDE)

- ▶ Video con esempio

Codice prodotto da Record and Play

```
driver.get(baseUrl + "/");
// enter to the insurance policy budget section
driver.findElement(By.id("linkautonoeq")).click();
// click on the second tab
driver.findElement(By.id("ext-gen24")).click();
// Bersani Compliant
driver.findElement(By.id("ext-gen286")).click();
driver.findElement(By.id("TBXXDPOXTargaVeicoloFam")).clear();
// type the plate number
driver.findElement(By.id("TBXXDPOXTargaVeicoloFam")).sendKeys("sa968ra");
// open the Bersani Class Menu
driver.findElement(By.id("ext-gen336")).click();
// select the class 2
driver.findElement(By.xpath("//div[@id='ext-gen382']/div[3]")).click();
// open the data picker
driver.findElement(By.id("ext-gen187")).click();
// click on today
driver.findElement(By.id("ext-gen435")).click();
// click on Avanti
driver.findElement(By.id("LBLXDPOXAvanti")).click();
```

Pro e contro

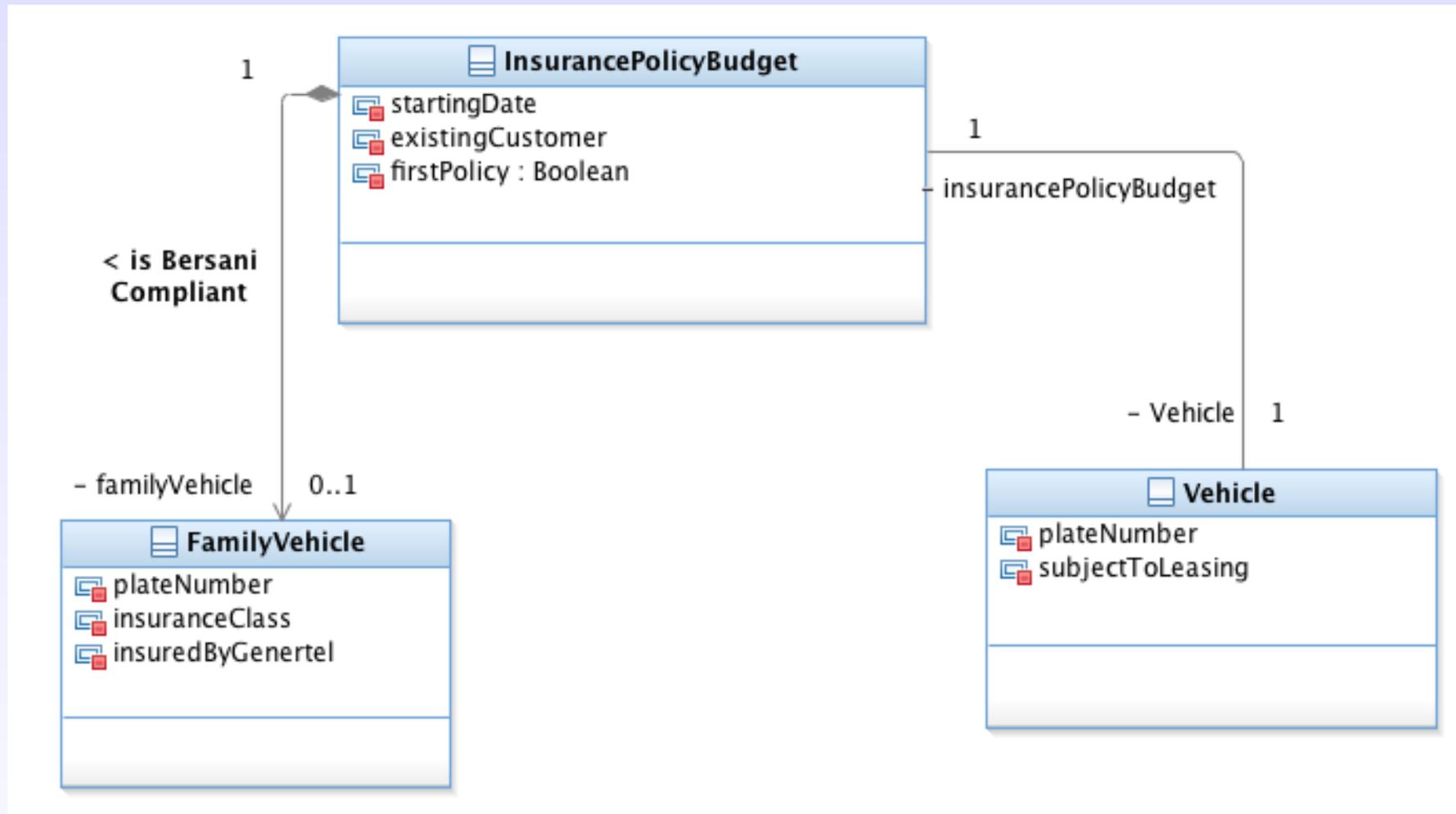
Pro

- ▶ rapido
- ▶ in grado di scoprire azioni mancanti

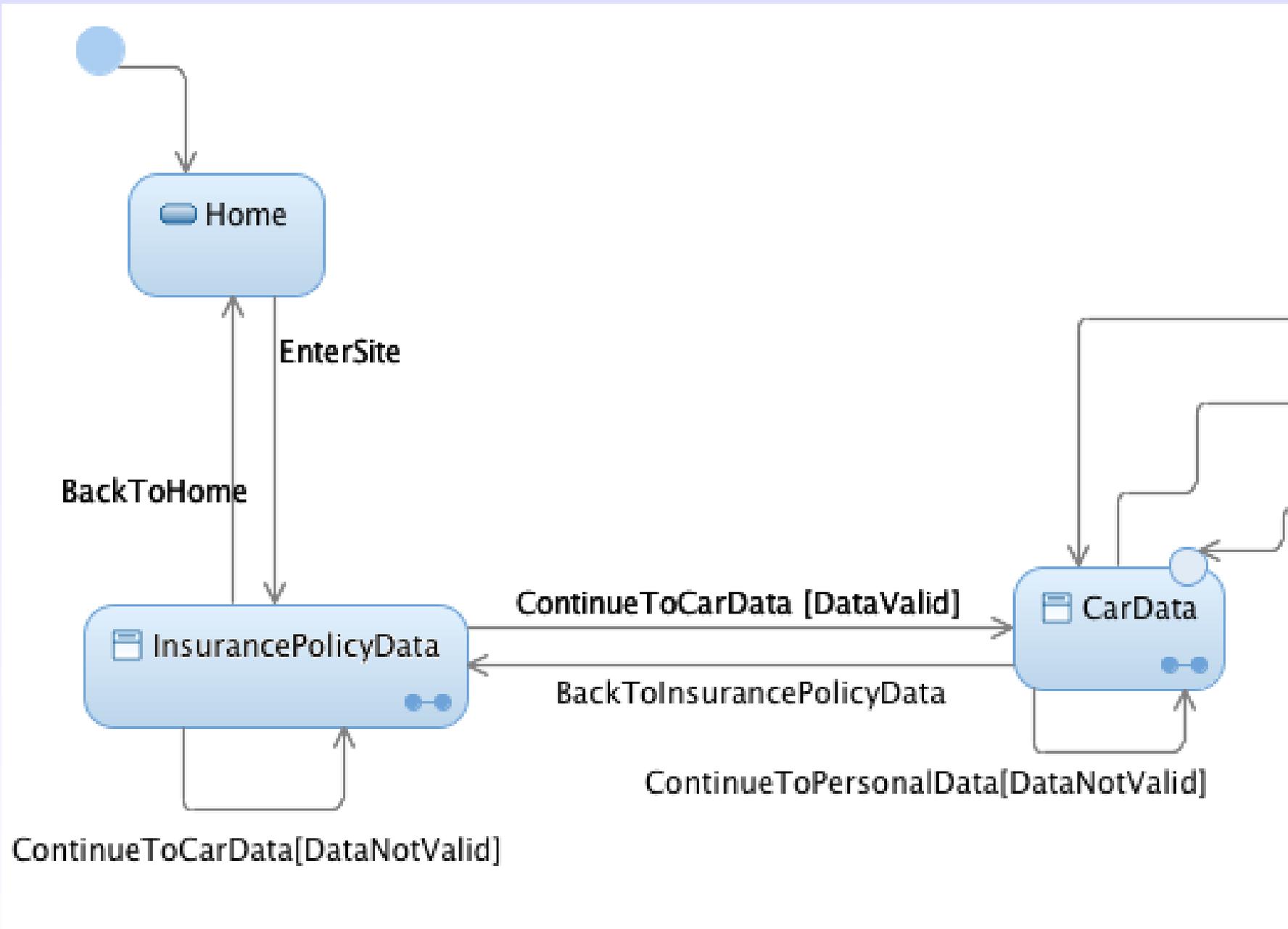
Contro

- ▶ codice molto fragile
- ▶ non ci sono asserzioni
- ▶ codice difficilmente comprensibile
- ▶ codice non parametrico rispetto ai dati
- ▶ codice non strutturato

Modello dei dati



Modello della dinamica



Annotazioni

per un'azione

```
<ann id="insuranceClassExists">  
  <for>chooseClass(class){bersaniClassMenuOpen}</for>  
  <locator type="xpath" parameter="value">  
    //div[@id='ext-gen382']/div/span[contains(text(),value)]  
  </locator>  
  <action>  
    click()  
  </action>  
</ann>
```

per uno stato

```
<ann id="continueToCarData">  
  <for>insurancePolicyData</for>  
  <locator type="getCurrentUrl">  
  </locator>  
  <assertion parameter="page">  
    <message>Page {page} not loaded</message>  
    <assertTrue>endsWith(page)</assertTrue>  
  </assertion>  
</ann>
```

Generazione automatica di infrastruttura di test

Test Harness

- ▶ compilatore che legge il modello e le annotazioni
- ▶ e produce codice sorgente del TH
- ▶ TH che isola dettagli della UI

```
genertelSUT.setConfiguration("default");  
genertelSUT.doAction("EnterSite");  
genertelSUT.doAction("NewInsuranceNewCar");  
genertelSUT.doAction("BersaniCompliant");  
genertelSUT.doAction("TypeNumberPlate", "SA968RA");  
genertelSUT.doAction("OpenBersaniClassAvailable");  
genertelSUT.doAction("ChooseClass", "2");  
genertelSUT.doAction("TypeDate", "01/01/2015");  
genertelSUT.doAction("Continue");  
genertelSUT.check("DataValid");
```

Benefici

1. poco sforzo di manutenzione: modifiche alle annotazioni e in qualche caso al modello
2. test case facili da comprendere
3. facili da scrivere da non specialisti
4. indipendenti da piattaforma
5. concisi: meno errori
6. possono contenere dati arbitrari
7. si possono scegliere varie configurazioni iniziali

Dati parametrici

- ▶ È possibile specificare dati in maniera parametrica e tabellare

```
<ann id="insurancePolicyBudget">
  <for>insurancePolicy</for>
  <data name="testInsurancePolicy" url="insurancePolicyBudget.csv">
</ann>
```

key	startingDate	existingCustomer	firstPolicy	valid
ip1	01/01/2015	no	true	yes
ip2	01/01/2000	yes	false	no
ip3	01/01/9999	no	true	no

```
DataRow dr = testInsurancePolicy.get("ip3");
genertelSUT.doAction("typing", dr.startingDate);
if (dr.valid.equals("no"))
  genertelSUT.checkActiveState("typingDate");
else
  genertelSUT.checkActiveState("chooseADate");
```

Test run logging

- ▶ log di tutto ciò che è stato sollecitato: pagina, elemento del DOM
- ▶ log di tutti gli esiti: pass/fail/error

```
1-Jun-2014 17.25.22 INFO: [TestFieldsFilled] SUT configured as "default"
...
1-Jun-2014 17.29.40 INFO: [TestFieldsFilled] Action EnterSite done (By.id="gen234_enter")
1-Jun-2014 17.29.40 INFO: [TestFieldsFilled] Entered in Home (By.id="get284_home")
1-Jun-2014 17.29.41 INFO: [TestFieldsFilled] Deactivated NotGenertelCustomer (By.cssLocator="ckbxCliente")
...
1-Jun-2014 17.29.52 INFO: [TestFieldsFilled] Checked DataValid (By.id="gen234_enter"): FAILED with date=01/01/99
```

Benefici:

- ▶ traccia esecuzione per verificarne il buon funzionamento
- ▶ traccia ad alto livello (facile da seguire)

Metriche di copertura

- ▶ che proporzione di stati/transizioni è stata testata
- ▶ cosa manca da testare
- ▶ grado di ridondanza in un test run

```
10-Jun-2014 17.32.45 INFO: [GenertelTests] This test suite covers
25/47 states (53\%); and 40/123 transitions (36\%).
```

```
10-Jun-2014 17.32.45 INFO: [GenertelTests] Tested transitions:
EnterSite (13/24),ContinueToCarData[DataValid] (8/24),
BackToInsurancePolicyData (2/24), ...
```

```
10-Jun-2014 17.33.15 INFO: [GenertelTests] Untested states: CarData,
NotGenertelCustomer, GenertelLogin, PersonalCode, GenertelCode,
GenertelCustomer, ...
```

```
10-Jun-2014 17.33.55 INFO: [GenertelTests] Untested transitions:
ConfirmGenertelCustomer\{NotGenertelCustomer\},
SubmitData\{GenertelLogin\}, AbortGenertelCustomer\{GenertelLogin\}, ...
```

Model-based testing

Consente di:

- ▶ ridurre sforzo di manutenzione
- ▶ eseguire i test case in automatico
- ▶ aumentare la qualità
- ▶ misurare livelli di copertura

Indice

Ingegneria dei requisiti e User-Centered Development (Giorgio)

Features e approcci agili (Andrea)

Design concettuale UI e usabilità (Giorgio)

Domain Driven Design (Andrea)

Model-driven acceptance testing (Giorgio)

Design for Testability (Andrea)

Conclusione

- ▶ Un approccio User-Centered Development è fondamentale
- ▶ UCD viene spalmato sulle varie fasi
- ▶ UCD richiede umiltà
- ▶ UCD sembra costoso, ma alla fine fa risparmiare (errori, tempo e denaro).
- ▶ UAT model-driven consentono di verificare qualità esterna senza incidere nel costo

Realizzare software usabile NON richiede dei superman



Contatti

Giorgio Brajnik

- ▶ brajnik@uniud.com
- ▶ 340.272 2173

Andrea Baruzzo

- ▶ abaruzzo@computer.org
- ▶ 339.395 5753

Palestre di ...

- ▶ **Usabilità**: 3-10 luglio: `www.friulinnovazione.it/eventi/palestra-di-usabilita`
- ▶ **Agile Project Management**: 9-16 ottobre: `www.friulinnovazione.it/eventi/palestra-di-agile-management`