

Разбиране на изискванията

■ Разбиране на изискванията (глава 5)

Slide Set to accompany

Software Engineering: A Practitioner's Approach, 8/e

by Roger S. Pressman

Лектор: Доц. д-р Ася Стоянова-Дойчева

Requirements Engineering-I

- **Начало (Inception)** — задават се набор от въпроси, които установяват:
 - основно разбиране на проблема
 - хората, които искат решение
 - естеството на решението, което се изисква
 - ефективността на предварителната комуникация и сътрудничество между клиента и разработчика
- **Извличане (Elicitation)** — извличане на изискванията от всички заинтересовани страни
- **Разработване (Elaboration)** — създаване на аналитичен модел, който идентифицира изискванията за данните, функциите и поведението
- **Преговори (Negotiation)** — съгласие за разработваната система, която е реалистична за разработчика и клиента.

Requirements Engineering-II

- **Спецификация (Specification)**— може да бъде една или повече от следните дейности:
 - Писане на документи
 - Множество от модели
 - Формални математически модели
 - Потребителски сценарии (use-cases)
 - Прототип
- **Валидация (Validation)**— механизъм за проверка, който търси:
 - грешки в съдържанието или интерпретацията
 - области, където може да се изисква изясняване
 - липсваща информация
 - противоречивост (основен проблем при разработка на големи системи)
 - несъвместими или нереалистични изисквания.
- **Управление на изискванията**

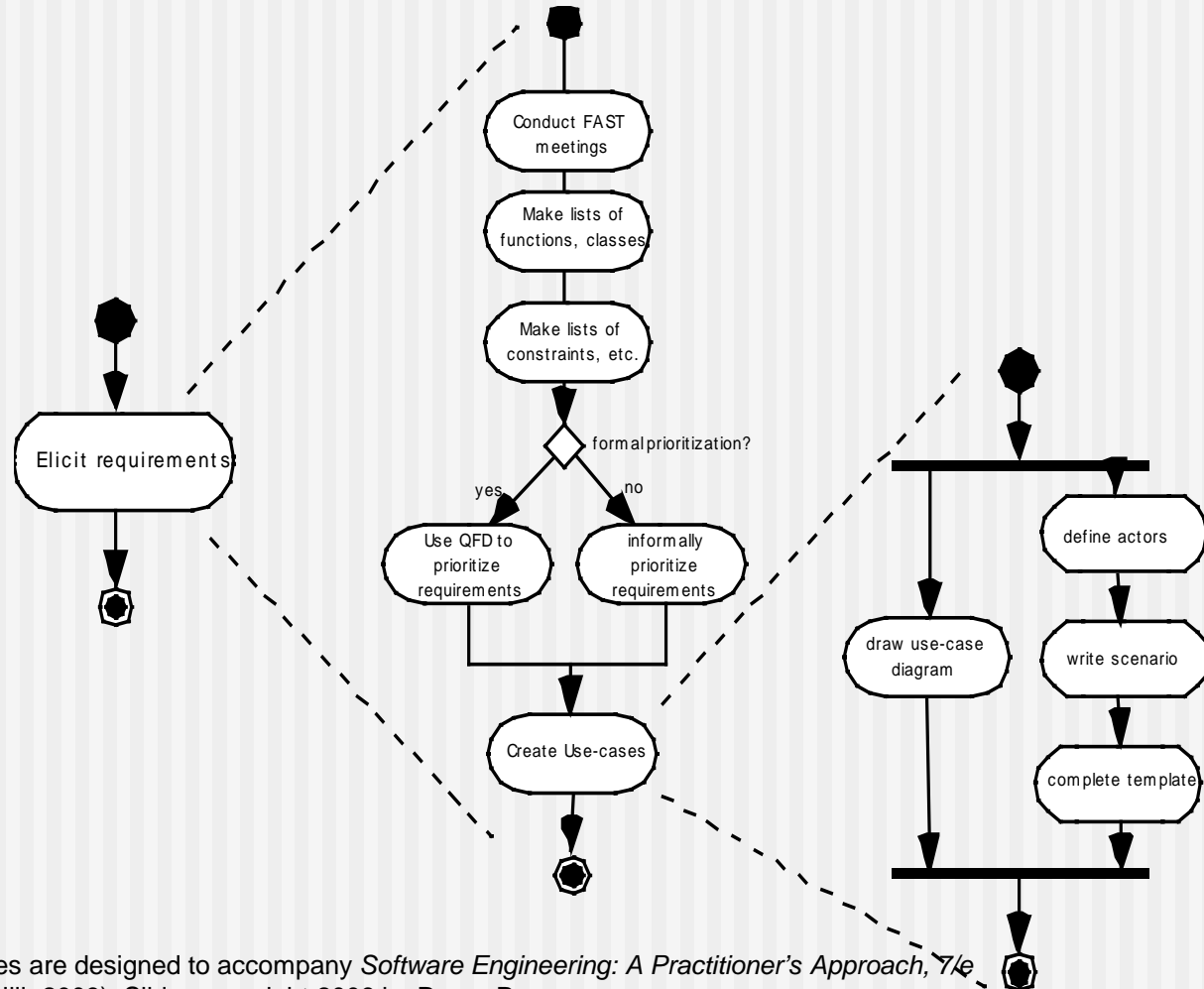
Начало

- Идентифициране на заинтересованите страни
 - С кой друг трябва да говоря?
- Разгледай различни гледни точки
- Работа в сътрудничество
- Първи въпроси:
 - Кой седи зад изискванията за тази работа?
 - Кой ще използва решението?
 - Какви ще бъдат икономическите ползи от успешно решение?
 - Има ли друг източник на решение?
- Нефункционални изисквания (NFR, QFD)
- Проследяемост на документите

Извличане на изискванията

- Провеждат се срещи (реални или виртуални) с участието на софтуерните инженери и клиентите
- Установени са правила за подготовка и участие
- Предложен е дневен ред
- модератор "facilitator" (може да бъде клиент, разработчик или външен) контролира срещата
- Използва се механизъм за дефиниране на изискванията (могат да бъдат работни листи, чат, стикери за стена, електронен бюлетин или виртуален форум)
- Целта е:
 - Да се идентифицира проблема
 - Да се предложат елементи от решението
 - Да се преговаря за различни подходи, и
 - Да се специфицира предварително множество от изисквания за решението.

Извличане на изискванията



These slides are designed to accompany *Software Engineering: A Practitioner's Approach, 7/e* (McGraw-Hill, 2009). Slides copyright 2009 by Roger Pressman.

Quality Function Deployment

- **Function deployment** определя как се възприема от клиента и каква стойност има за него всяка функция на исканата системата
- **Information deployment** идентифицира данните и събитията
- **Task deployment** разглежда поведението на системата
- **Value analysis** определя относително приоритета на изискванията

Резултат от Извличането

- документ за необходимост и приложимост
- документ за ограничение на системата или продукта
- списък на клиенти, потребители и други заинтересовани, които са участвали в извличане на изискванията
- описание на техническата среда на системата
- списък на изискванията и ограниченията на домейна приложени върху тях
- набор от сценарии за ползване, които осигуряват поглед върху използването на системата или продукта при различни условия на работа.
- разработени прототипи за по-добро дефиниране на изискванията

Извличане на изисквания при Agile

- Клиентите създават user stories.
 - Просто изискване към системата от гледна точка на потребителя;
 - Записани на малки бележки (карти) на езика на потребителя
 - Обикновено липсват общи бизнес цели и нефункционални изисквания

Разработване на аналитичен модел

- Елементи на аналитичния модел
 - Базиран на сценарии елементи (Scenario-based)
 - Functional— преработени разкази за софтуерни функции
 - Use-case— описание на взаимодействие между “actor” и системата
 - Базиран на класове елементи (Class-based)
 - Базиран на сценарии
 - Елементи на поведени (Behavioral)
 - Крайни автомати (State diagram)
 - Елементи ориентирани към потоци (Flow-oriented)
 - Диаграми на потока от данни (Data flow diagram)

Use-Cases

- Множество от потребителски сценарии, които описват използването на системата
- Всеки сценарии е описан от гледна точка на актьор – човек или устройство което си взаимодейства със системата по някакъв начин
- Всеки сценарии отговаря на следните въпроси:
 - Кой е основен актьор и кой вторичен?
 - Какви са целите на актьорите?
 - Какви предусловия трябва да съществуват, преди стартиране на Use case?
 - Какви основни задачи или функции се изпълняват от актьора?
 - Какви разширения могат да бъдат разгледани, след описание на use case?
 - Какви вариации на взаимодействие на актьора са възможни?
 - Каква системна информация актьора ще усвои, създаде или промени?
 - Ще трябва ли актьора да информира системата за промени във външната среда?
 - Каква информация актьора иска да получи от системата?
 - Желает ли актьора да бъде информиран за неочаквани промени?

SafeHome System- изисквания

- Дефиниране на актьорите: **homeowner** (a user), **setup manager** (същия човек като homeowner, но играе друга роля), **sensors** (устройства свързани към системата), и **подсистема за наблюдение и отговор** (централа, която наблюдава SafeHome системата за охрана). За примера ще разгледаме само homeowner актьора. Той си взаимодейства с функциите на системата по различен начин като използва контролни панел на алармата или компютър:

SafeHome System- изисквания

- Въвежда парола, за да се позволят други взаимодействия.
- Пита за статуса на пазена зона.
- Пита за статуса на сензор.
- Натиска паник бутона в извънредна ситуация.
- Активира/ Деактивира системата.

- Homeowner наблюдава SafeHome контролния панел, за да определи дали системата е готова за въвеждане. Ако системата не е готова се показва съобщение на LCD екрана и homeowner трябва физически да затвори врата или прозорец.
- Homeowner използва клавиатурата за въвеждане на парола. Паролата се сравнява с валидна записана в системата. Ако паролата е невалидна панела ще изписка веднъж и ще даде възможност за ново въвеждане. Ако паролата е коректна контролния панел ще чака за следващо действие
- 3. Homeowner избира „stay“ или „away“ за да активира алармата. „Stay“ активира само сензорите за периметър (вътрешните са деактивирани). „Away“ активира всички сензори.
- 4. Когато алармата е активирана светва червена лампа на контролния панел.

Use case: InitiateMonitoring

Primary actor: Homeowner.

Goal in context: To set the system to monitor sensors when the homeowner leaves the house or remains inside.

Preconditions: System has been programmed for a password and to recognize various sensors.

Trigger: The homeowner decides to “set” the system, i.e., to turn on the alarm functions. Scenario:

1. Homeowner: observes control panel
2. Homeowner: enters password
3. Homeowner: selects “stay” or “away”
4. Homeowner: observes read alarm light to indicate that SafeHome has been armed

Exceptions:

1. Control panel is not ready: homeowner checks all sensors to determine which are open; closes them.
2. Password is incorrect (control panel beeps once): homeowner reenters correct password.
3. Password not recognized: monitoring and response subsystem must be contacted to reprogram password.
4. Stay is selected: control panel beeps twice and a stay light is lit; perimeter sensors are activated.
5. Away is selected: control panel beeps three times and an away light is lit; all sensors are activated.

Priority: Essential, must be implemented

When available: First increment

Frequency of use: Many times per day

Channel to actor: Via control panel interface

Secondary actors: Support technician, sensors

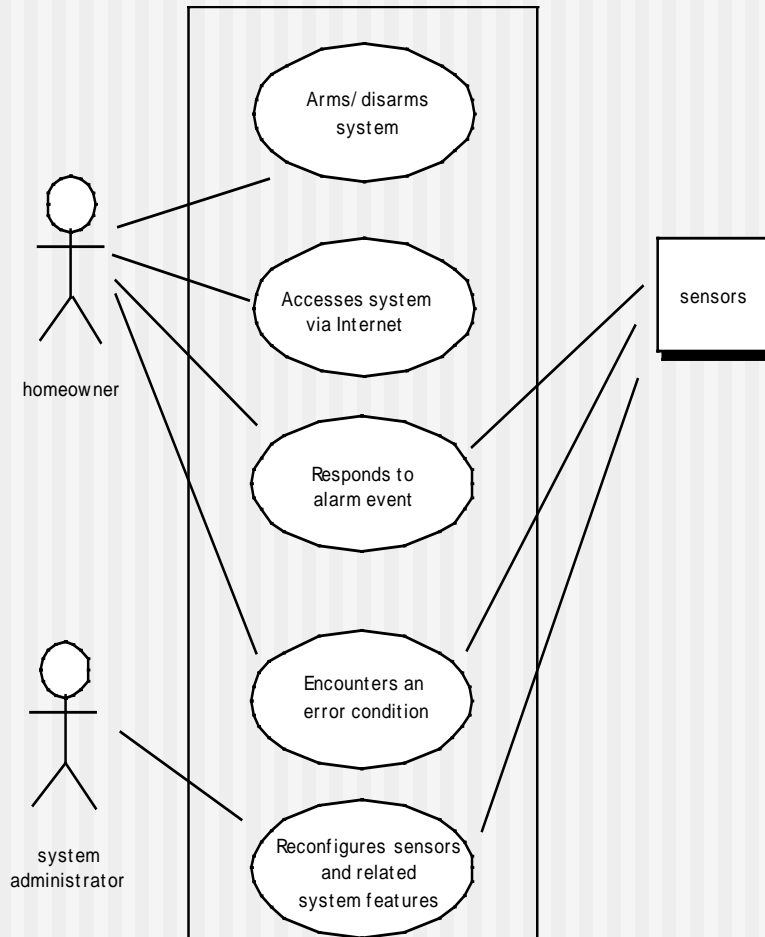
Channels to secondary actors:

Support technician: phone line Sensors: hardwired and radio frequency interfaces

Open issues:

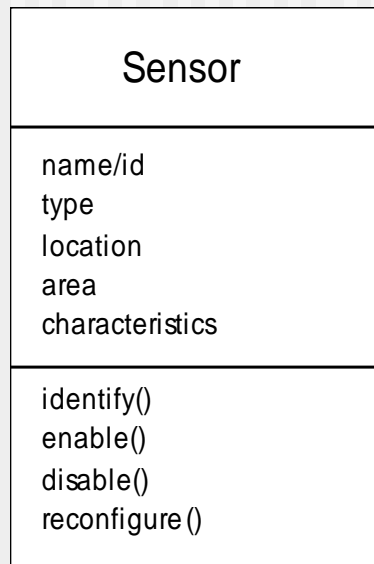
1. Should there be a way to activate the system without the use of a password or with an abbreviated password?
2. Should the control panel display additional text messages?
3. How much time does the homeowner have to enter the password from the time the first key is pressed?
4. Is there a way to deactivate the system before it actually activates?

Use-Case диаграмма

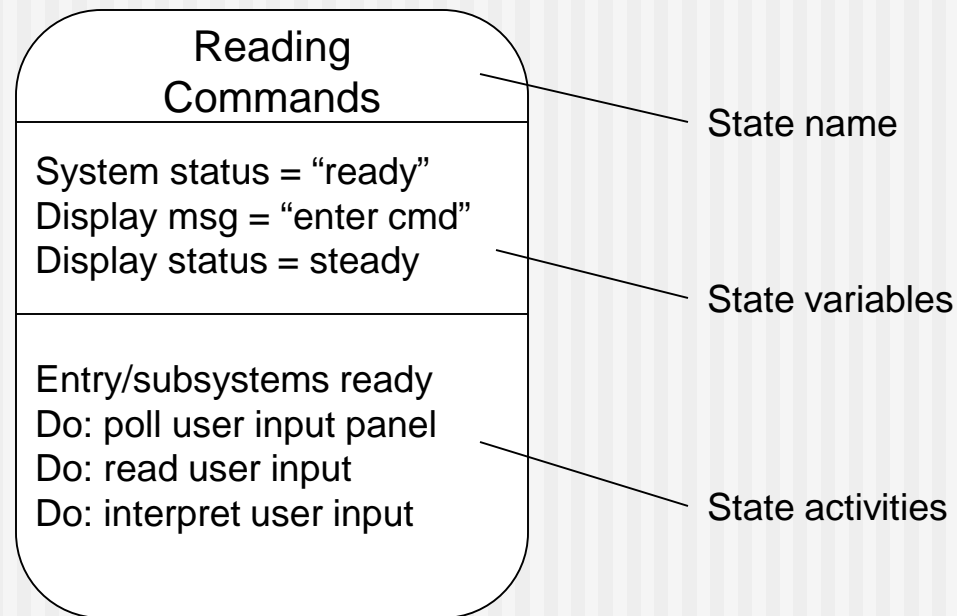


Class диаграма

От *SafeHome* система ...



State диаграма



Шаблони за анализ

Име на шаблона: Descriptor, който описва шаблона.

Цел: Описва какво представя шаблона.

Мотивация: Сценарии, който илюстрира как може да бъде използван шаблона за конкретен проблем.

Контекст: Описание на външно влияние, което може да повлияе върху използването на шаблона и също външни влияния, които ще бъдат решени при прилагането му.

Решение: Описание на начина на прилагане на шаблона за решаване на проблема с акцент върху структурните и поведенчески проблеми.

Последиствия: Какво се случва, когато шаблона е приложен и какви компромиси съществуват по време на прилагането.

Проект: Как шаблона може да се реализира, чрез използване на проектни шаблони.

Приложения на шаблона: Примери за приложение на шаблона.

Свързани шаблони: един или повече шаблони за анализ свързани с този.

Изисквания за договаряне

- Идентифициране на ключовите заинтересовани лица.
- Определяне за всеки заинтересован “win conditions”
 - Win conditions не са винаги очевидни.
- Преговарям
 - Работи за множество от изисквания, които водят до “win-win”

Изисквания за валидация- I

- Отговаря ли всяко изискване на общата цел на системата/продукта?
- Специфицирани ли са всички изисквания на подходящо ниво на абстракция?
- Изискването реално необходимо ли е или представя допълнителна възможност, която не е в общата цел на системата?
- Дали всяко изискване е ограничено и недвусмислено?
- Дали всяко изискване е прието?
- Дали някое изискване е в конфликт с други изисквания?

Изисквания за валидация- II

- Дали всяко изискване е постижимо в изискваната техническата среда на продукта?
- Дали всяко изискване може да бъде тествано след имплементацията?
- Дали моделът на изискванията правилно отразява информацията, функцията и поведението на системата, която ще се разработва?
- Бил ли е моделът на изискванията разделен на части по начин, който представя прогресивно по- детайлна информация за системата?
- Използвани ли са шаблони за анализ, които да опростят модела на изискванията? Валидирани ли са всички шаблони? Съвместими ли са шаблоните с изискванията на клиента?