

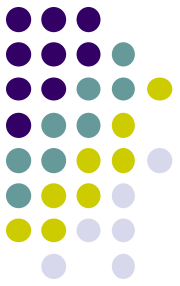
Инженеринг на изискванията

Лекция 1

Съдържание



- Основни концепции и дефиниции
- Роля на Инженеринга на Изискванията (ИИ) като част от системния инженеринг
- Какво инженерите по изискванията правят?
- Връзки на ИИ с
 - организационния процес
 - процеса на разработката



Какво е **изискване** на софтуерна система?

What is a software requirement?

Изискване на софтуерна система - 1



IEEE Stud. 610.12-1990

Дефиниция (софтуерно изискване):

- 1) Условие или способност на софтуера, необходимо на потребителя, за да реши проблем или да постигне определена цел.
- 2) Условие или способност, което трябва да има системата или компонент на системата, за да удовлетвори договор, стандарт, спецификация или друг формално наложен документ.
- 3) Документално представяне на условието или способността от т.1) и т.2) .

/Клаус Пол/

Изискване на софтуерна система - 2



- Описанието на **услугите/функционалностите** и на **ограниченията** на системата са изисквания на системата.
- Документирано представяне на **нужда, способност или качество** на софтуерна система.
- Изискванията се генерират по време на процеса на инженеринга на изискванията.

Процес на инженеринг на изискванията (Requirements Engineering process) -1

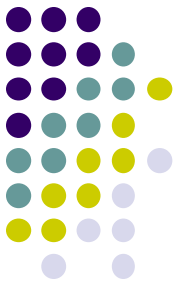


Дефиниция: Систематичен процес на идентифициране, анализиране, документиране и проверка/валидиране на функционалностите (услугите) и ограниченията на даден софтуер.

Ian Sommerville

Процес на установяване и документиране на услугите, които *клиент изисква* от системата, както и на *ограниченията*, при които системата да работи и да бъде разработвана.

Инженеринг на Изискванията - класическото понятие



- *Дефиниция.* Инженеринг на изискванията (ИИ): Прилагане на систематичен, дисциплиниран, количествено измерим подход към спецификацията и управлението на изискванията; приложение на инженерството към изискванията.
- *Метафорично:* предварително инженерство
- *Цел:* **пълни, недвусмислени** изисквания преди проектирането
 - Спецификацията на идеалните изисквания трябва да описва бъдещата система (или системата за промяна) толкова пълно и точно, че разработчиците да могат да внедрят системата, без да искат обратна информация.

However

Smells: paper, process

Reality check: Does this always work? ⁷

Къде е мястото на ИИ в цялостния процес на разработка на софтуер?



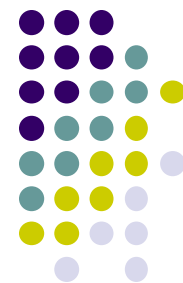
Процес на разработка на софтуерния продукт

„Прост“ пример:

A ski resort operates several chairlifts. Skiers buy RFID-equipped day access cards. Access to the lifts is controlled by RFID-enabled turnstiles. Whenever a turnstile senses a valid access card, it unlocks the turnstile for one turn, so that the skier can pass.



ИИ (RE) – същност 1:

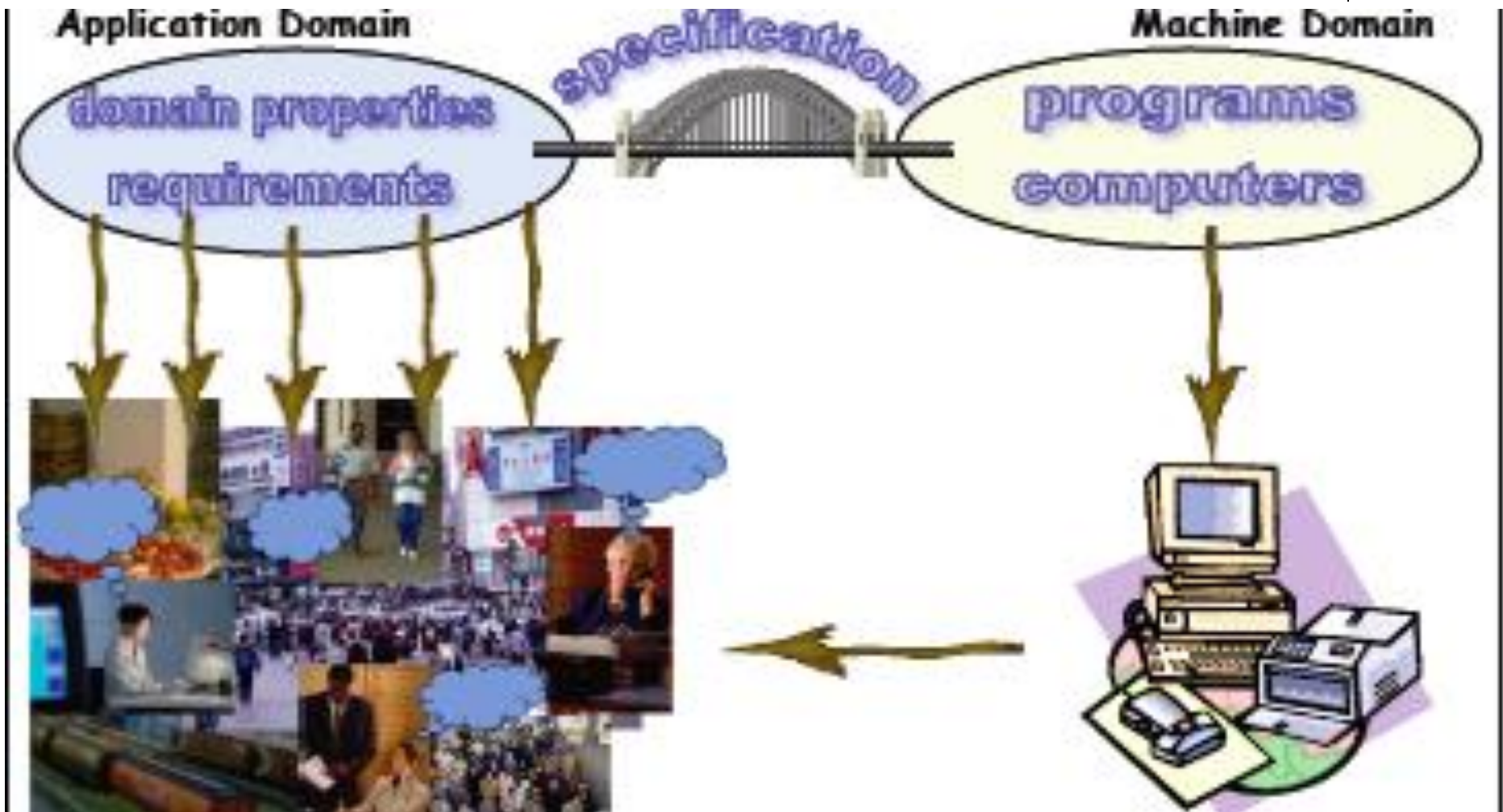
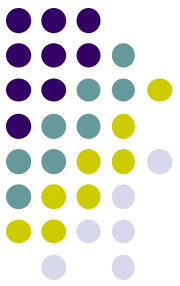


RE is a set of activities concerned with identifying the purpose of a software system, and the contexts in which it will be used. Hence, **RE acts as the *bridge* between the real world needs of users, customers, and other constituencies affected by a software system**, and the capabilities and opportunities afforded by *software-intensive technologies*.

Steve Easterbrook

ИИ е в ролята на **мост** между нуждите на клиентите, ползвателите и останалите заинтересовани от работата на създаваната софтуерна система и техническите възможности, предлагани от (интензивните) софтуерните технологии.

Предизвикателствата?



ИИ – същност 2



Not a phase
or stage

Communication
is important as
analysis

Quality means
**fitness for
purpose.**

Cannot say
anything about
quality unless you
understand the
purpose.

Requirements Engineering (RE) is a set of activities concerned with identifying and communicating the purpose of a software-intensive system, and the contexts in which it will be used. RE acts as the bridge between the real-world needs of users, customers, and other constituencies affected by a software system, and the capabilities and opportunities afforded by software-intensive technologies.

Designers need to know **how and where** the system will be used.

Requirements are **partly about what we need ...**

... and partly about what is possible.

Need to identify **all the stockholders** not just the customer and user.

Why it is “requirements” ?

Why it is “engineering” ?

Software-intensive systems – 1

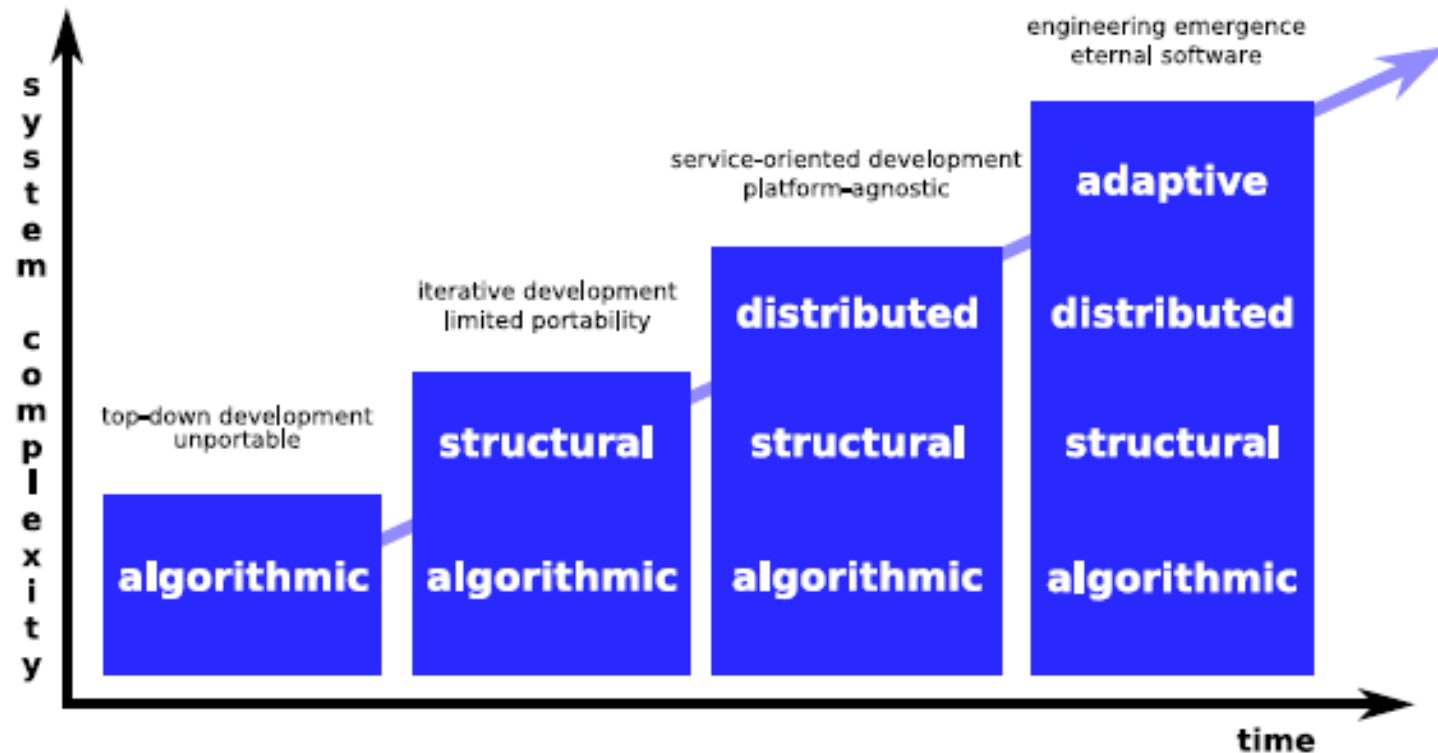


- Software (on its own) is useless
 - Software is an abstract description of a set of computations
 - Software only becomes useful when run on some hardware
 - Software + Hardware = Computing system
- A computer system (on its own) is useless
 - Only useful in the context of some *human activity* that it can support
 - A new computer system will change human activities in significant ways
- **Software + Hardware + Human activities = Software-intensive system**

Software-intensive systems 2



- Software makes many challenges during the time and today:
 - It is complex and adaptable
 - It can be rapidly changed on-the-fly
 - It turns general-purpose hardware into a huge variety of useful machines



Изискванията се откриват, а не намират - 1

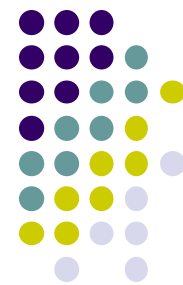


*“Requirements **cannot be** observed or asked for from the users, but **have to be created together** with all the stakeholders.”*

Vesa Torvinen

- So far we used words like requirements:
 - capture (обхващам)
 - gathering (събирам в сбирка, колекция ...)
 - trawling (тралене)
 - elicitation (извличане, изваждане, разкриване)

Изискванията се откриват, а не намират - 2



- “Откриване” означава много и различни дейности:
 - Нови идеи (being open to new ideas);
 - Креативност (applying creative effort);
 - Работа в екип (working as a team);
 - Фокусирани въпроси (asking questions that focus the search);
 - Да се намери специфичното (intending to find particular kinds of thing);
 - Да се представи в разумна рамка на разглеждане (fitting whatever is found into a reasoned framework);
 - Отнасяне към подобни случаи (relating whatever is found to similar discoveries).
- Примери!



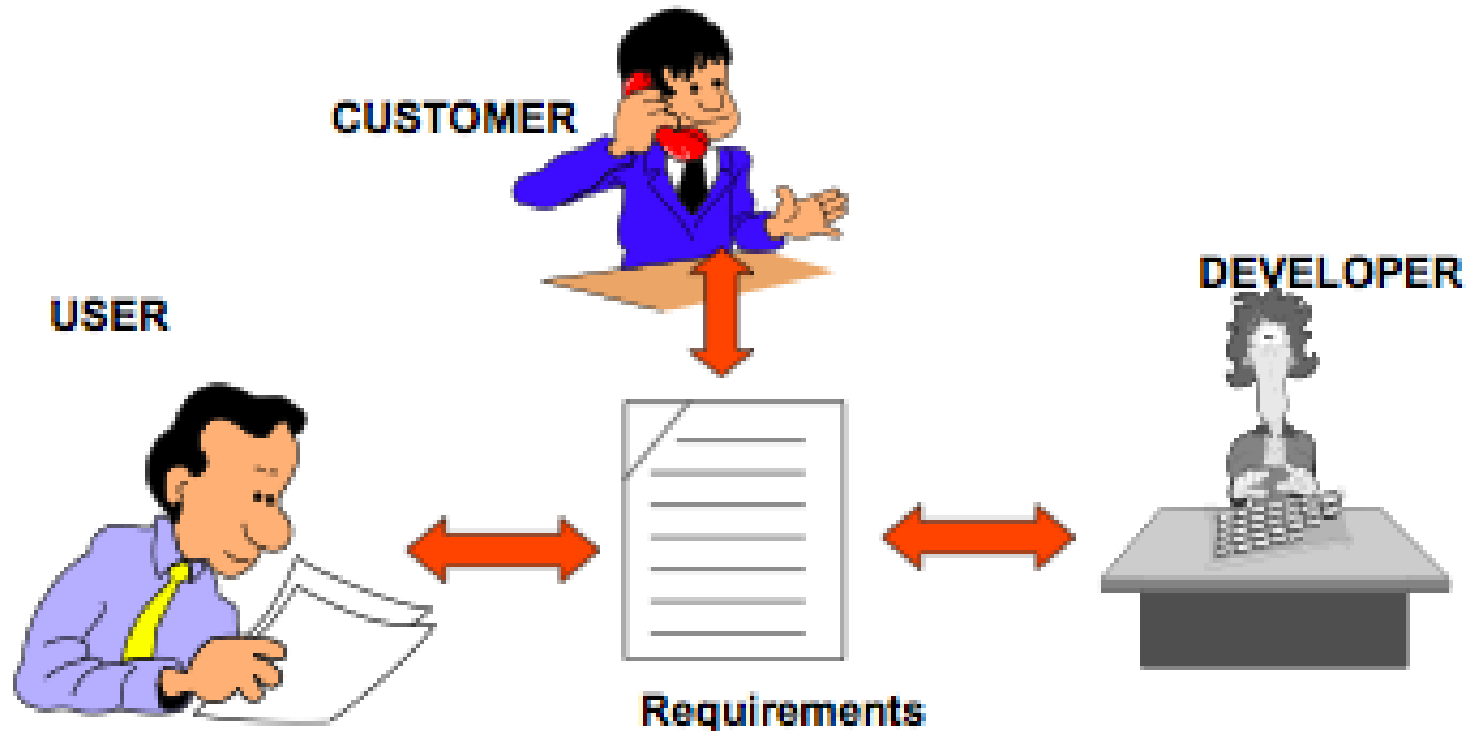
Когато изграждаме софтуерна система:

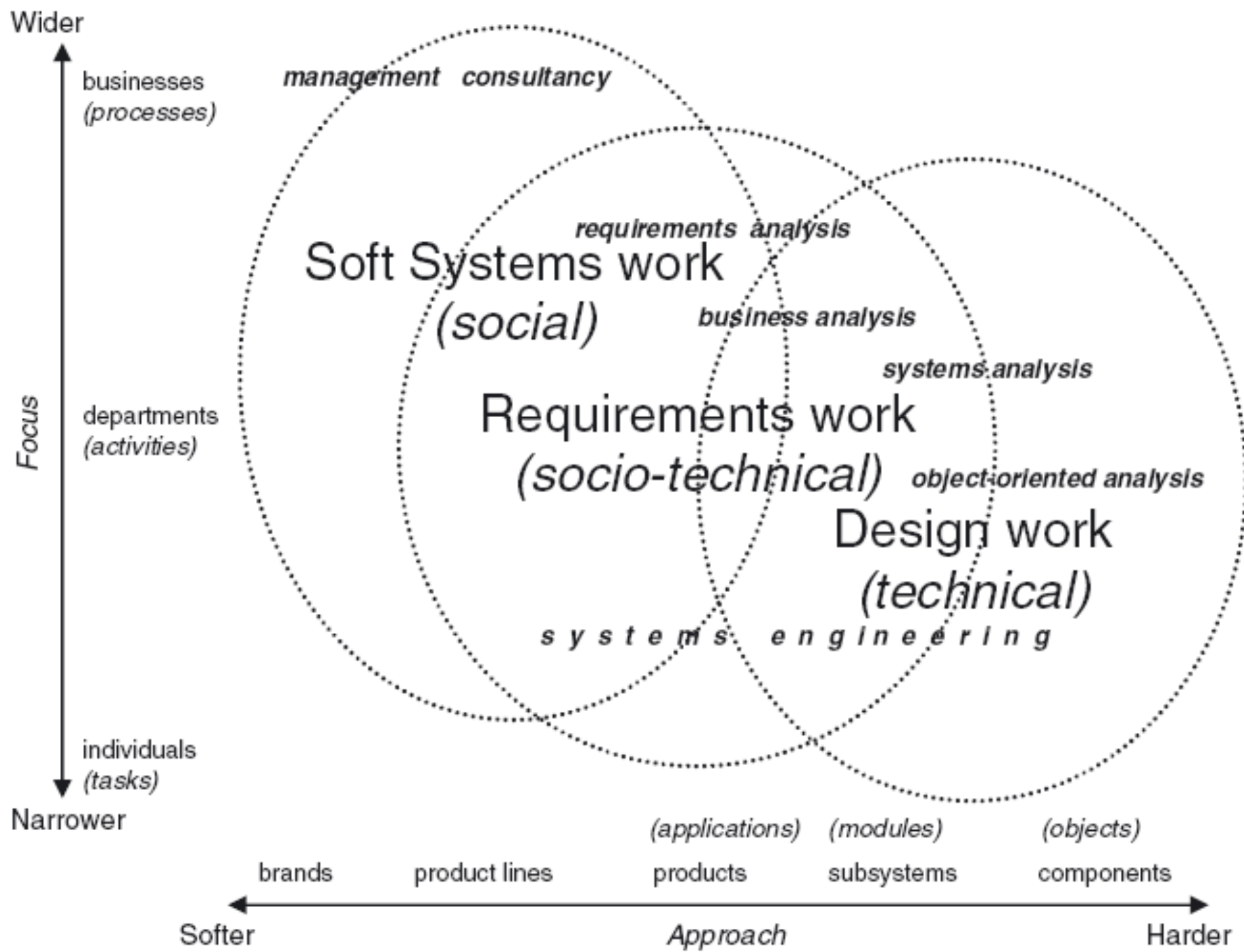
- How do we determine the requirements?
- How can we analyse and document these requirements?
- How do we make sure that we've got the right requirements?
- How do we manage and evolve the requirements?

... и още защо са важни изискванията?



Те са единствената част, която всички разбират.
Основа за бъдещата разработка.







**Какво е спецификация на
изискванията?**

What is a requirement specification?

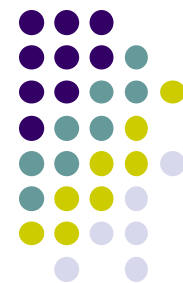
Спецификация на изискванията - 1



- The set of all requirements forms the basis for subsequent development of the system or system component.

Isn't it all?

Спецификация на изискванията - 2



Спецификацията е повече от списък ...

- Това е **мрежа от свързани, зависими елементи на изискванията**, включващи дефиниции, цели, обосновки, измервания и друга информация
- ‘The requirements’ in the broad sense means a network of *interrelated requirement elements*: a requirement that satisfies a goal, is justified in a **rationale model**, using terms defined in the project dictionary, etc.

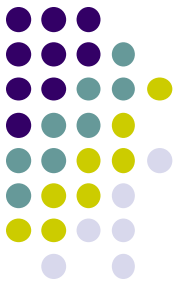
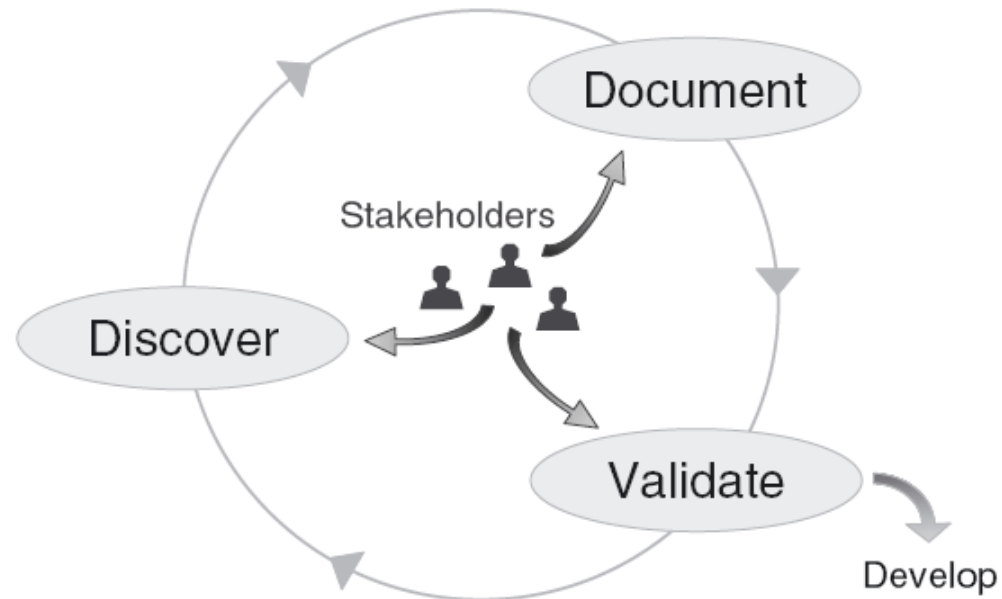
Discovery as search



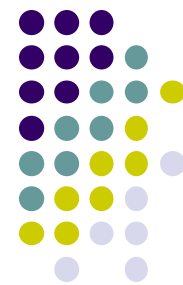
- Начин на мислене (A good way of thinking about discovery is as a search).
- Структуриране на знанието. (The structure of what you know drives what you discover next).
- **Колкото по-добре е организирано знанието за изискванията, толкова по-добре ще може да се открие онова, което наистина е необходимо.**
- Например:
 - Формулирай целта (на системата, функционалността, услугата).
 - Използвай сценарии, за да изследваш как можеш да постигнеш целта.
 - Търси изключенията, който може да се срещнат.

Цикъл на откриването

- A generic discovery process can be drawn as a simple inquiry cycle
- An inquiry cycle is more or less what it says:
 - (Цикъл от) дейности в екип за формулиране на решение
 - Проверка на ефективността на решението



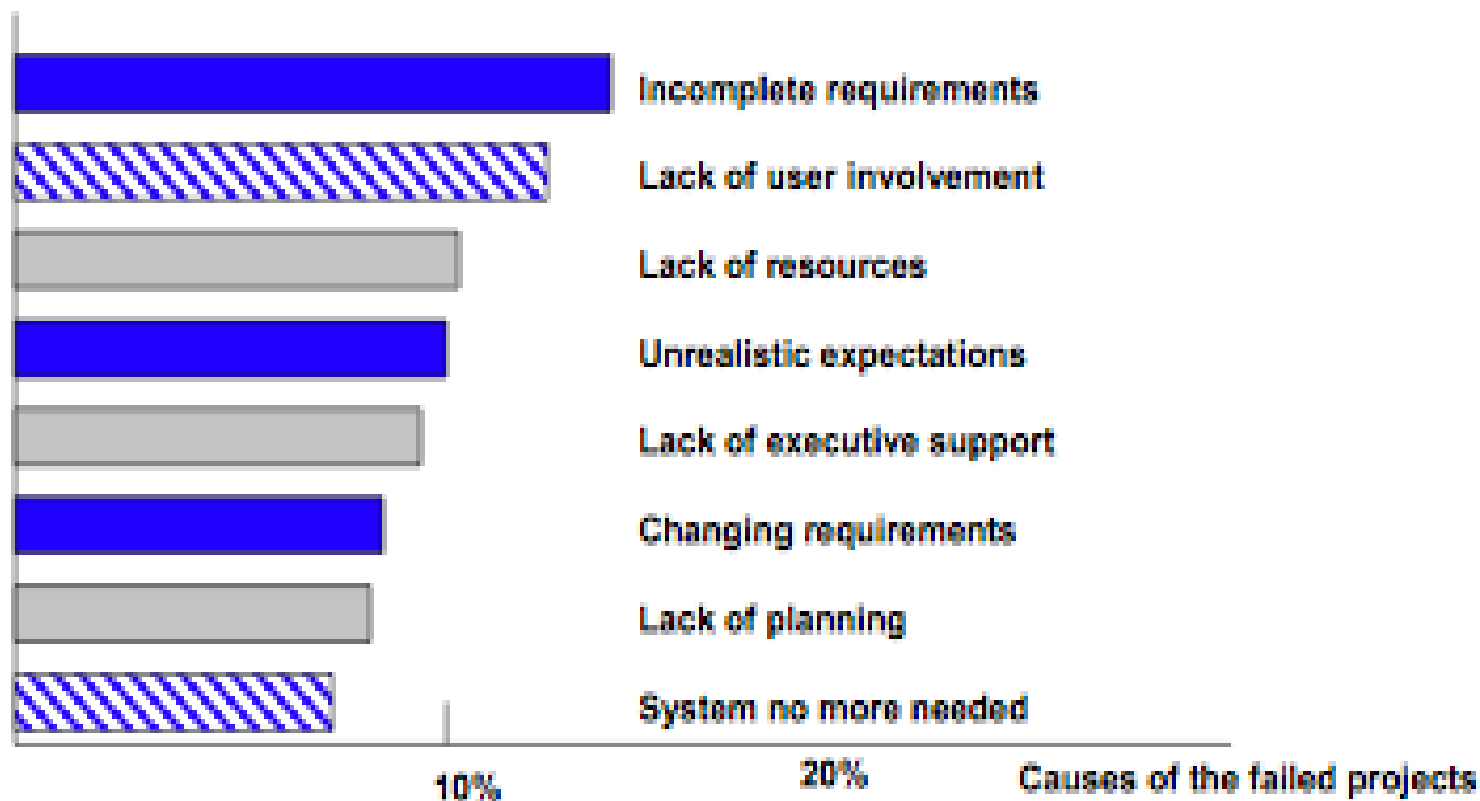
Защо изискванията са (много) важни?



- Колко струва ИИ? *About 15% of system development costs?*

НО !

- *Непълните и неконсистентни изисквания* са най-честите причини за проблеми на системата.

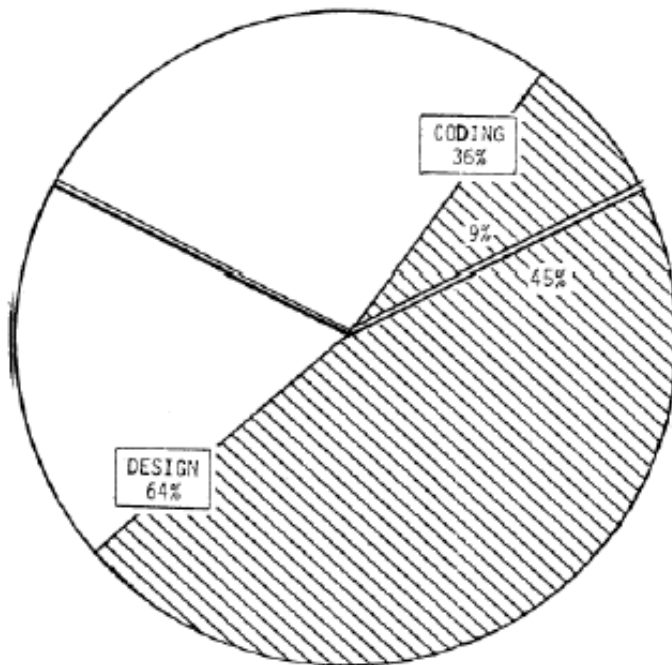


Мотивация за ИИ - 1



- “Errors are more frequent during requirements and design activities and are the more expensive the later they are removed”

Wording by [Endres2003]



ERRORS FOUND
DURING OR AFTER
ACCEPTANCE TEST

Design: Errors fixed by changing the design
Coding: Errors fixed by changing only the code

From [Boehm 1975]



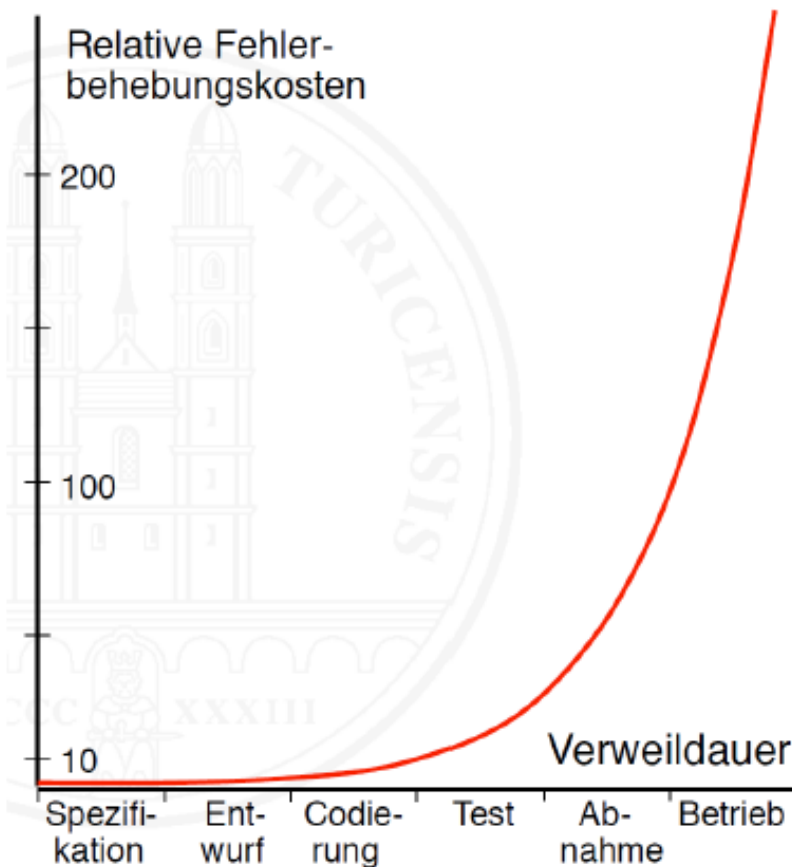
Barry Boehm

[Image: Wikipedia]

Мотивация за ИИ - 2



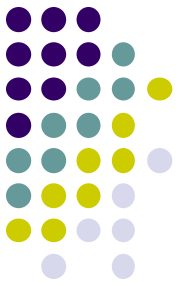
“Fix it later” is expensive



- Cost for removing errors depends on how long it stays in the software
- The later errors are detected, the more expensive their removal is
 - E.g. due to required re-design and implementation
- Good RE avoids requirements errors and thus saves cost for removing errors later

Мотивация за ИИ - 3

Importance of requirements engineering



Barry Boehm investigated the cost to fix errors in the development of large software systems.

- Causes of
 - **Top three success factors are:**
 - user involvement;
 - executive management support;
 - a clear statement of requirements.

The top three factors leading to failure were:

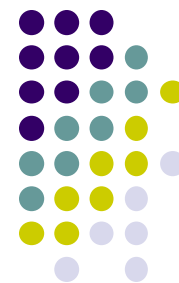
- *lack* of user input;
- *incomplete requirements* and specifications;
- *changing* requirements and specifications.

Some observations about RE



- Perfecting a specification may not be cost-effective
 - Requirements analysis has a cost – (**how much?**)
 - For different projects the cost-benefit *balance* will be different
- The problem statement should *never* be treated as fixed
 - Change is inevitable, and therefore must be planned
 - There should be a way of incorporating changes periodically

Каква е връзката между изисквания и проектирането?



- Два основни (*парадоксални* 😊) принципа:

“Requirements and design are **interleaved**. They should, ideally, be separate processes but in practice this is impossible.”

Somerville

- Полезно е да **разделим** проблема от решението
 - и да документира проблема отделно от всички дизайнерски решения

НО!

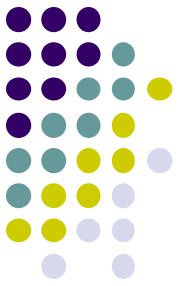
- Това разделяне **никога не може да се постигне** в пълна степен
 - защото дизайнът променя света и следователно променя първоначалния проблем 😊.

Systems Engineering vs. Software Engineering



Инженеринга на изискванията се прилага както и към софтуерни системи, така и към системите изобщо.

- В софтуерното инженерство обикновено се приема, че начинът, по който софтуерът взаимодейства със света, е фиксиран, като се използват стандартизирани входни и изходни устройства.
- Задачата в системното инженерство е да се проектира цяла система, от която софтуерът е само един компонент.



What do requirements engineers do?

Какво правят инженерите по изискванията (аналитиците) ? - 1



- Изявяват се в началото на проекта
 - Идентифицират проблем, който се нуждае от решение:
 - Недоволството от текущото състояние
 - Нова възможност за бизнес
 - Потенциал за спестяване на разходи
 - *Анализът* на изисквания е агент на промяната

Какво правят инженерите по изискванията?

-2



- Инженерите по изискванията трябва да:
 - Идентифицират проблема/възможността
 - Which problem needs to be solved? (identify problem boundaries)
 - Where is the problem? (understand the Context/Problem domain)
 - Whose problem is it? (identify stakeholders)
 - Why does it need solving? (identify the stakeholders' goals)
 - How might a software system help? (collect some scenarios)
 - When does it need solving? (identify development constraints)
 - What might prevent solving it? (identify feasibility and risk)
 - Да бъдат/станат експерти в проблемната област

Какво правят инженерите по изискванията?

– взаимоотношения 1



- **С други субекти на организацията на процеса**
 - Product management
 - Marketing
 - Customer relationship management
- **Фокусът е в дейности по търсене и записване на информация :**
 - in RE process of making sense of the answers, requirements engineers may *build models, and test them* in various ways.
 - They *will communicate and negotiate* with a variety of stakeholders, to reach agreement on the nature of the problem.
 - They will help to *bridge* the gap between the problem and the solution,
 - and *manage the evolution* that takes place as the problem changes.

взаимоотношения 2



- С други субекти на разработването
 - Project management
 - Design
 - Quality assurance
 - System maintenance

Simple but not easy



- Requirements work is simple but not easy:
it is a *craft*, which requires skills.
- ... Fortunately, that can be learned !



■ Lots of sources for today's terminology

- Textbooks and articles about RE
- IEEE 610.12 (1990) – a slightly aged glossary of software engineering terminology
- IEEE 830-1998 – an outdated, but still cited RE standard
- ISO/IEC/IEEE 29148 (2011) – a new, but still rather unknown RE standard; provides definitions of selected terms, some of them being rather uncommon
- IREB Glossary [Glinz 2013] – influential through IREB's certification activities; used as a terminology basis in this course



FAQS about requirements

- What are requirements?

- What is requirements engineering?

- How much does requirements engineering cost?

- What is a requirements engineering process?



FAQs contd.

- What happens when the requirements are wrong?

- What is a requirements document?

- What are system stakeholders?

- What is the relationship between requirements and design?