



СОФИЙСКИ УНИВЕРСИТЕТ “СВ. КЛИМЕНТ ОХРИДСКИ”

Факултет по математика и информатика

Специалност: Информатика

Магистърска Програма: Вградени Системи

Преподавател: д-р Йоаннис ПАТИАС

ПРОЕКТИРАНЕ НА РОБОТИЗИРАНИ СИСТЕМИ

Дисциплина:

Проектиране на роботизирани системи



E-mail:

ioannis.patias@gmail.com

Теми на лекциите

1. Основни понятия
2. Процеса на проектирането
3. **Провеждане на експерименти и тестове**
4. Прилагане на оптимизация
5. Осигуряване на стабилност
6. Захранване
7. Навигационни сензори
8. Системи в дискретно време
9. Контрол на обратната връзка
10. Проектиране на контрола на системата

Теми на лекциите

1. Основни понятия
2. Процеса на проектирането
3. **Провеждане на експерименти и тестове**
 1. Идеята за експерименти и тестове
 2. физикъл компютинг (physical computing)
 3. Физикъл компютинг срещу роботика
 4. Инсценериране на поведения (sketching)
 5. Взаимодействие
 6. Проектиране на взаимодействие
 7. Прототип
 8. Предимства и недостатъци
 9. Прототайпинг (prototyping)
 10. Ардуино прототайпинг
 11. Ардуино прототайпинг щит

Идеята за експерименти и тестове

- експериментите и тестовете, разширяват понятието изчислителна техника
- минаваме от интеракцията с монитора в реалния-физическия свят
- вградената система, чрез сензори, задвижващи устройства (actuators) и индикатори създава устройства, инсталации дори цели среди за интеракция
- така изчислителното взаимодействие (computational interaction) излиза от всякакви клишета

Идеята за експерименти и тестове

- фокуса по този начин се премества от машината върху човека
- т.е. трябва да се съобразяваме с това как човека участва, и/или как разбира дадена физическа среда
- с други думи, чрез експерименти и тестове проучваме поведения:
 - реактивни,
 - изразителни,
 - интерактивни и
 - въплътени (embodied)

Физикъл компютинг

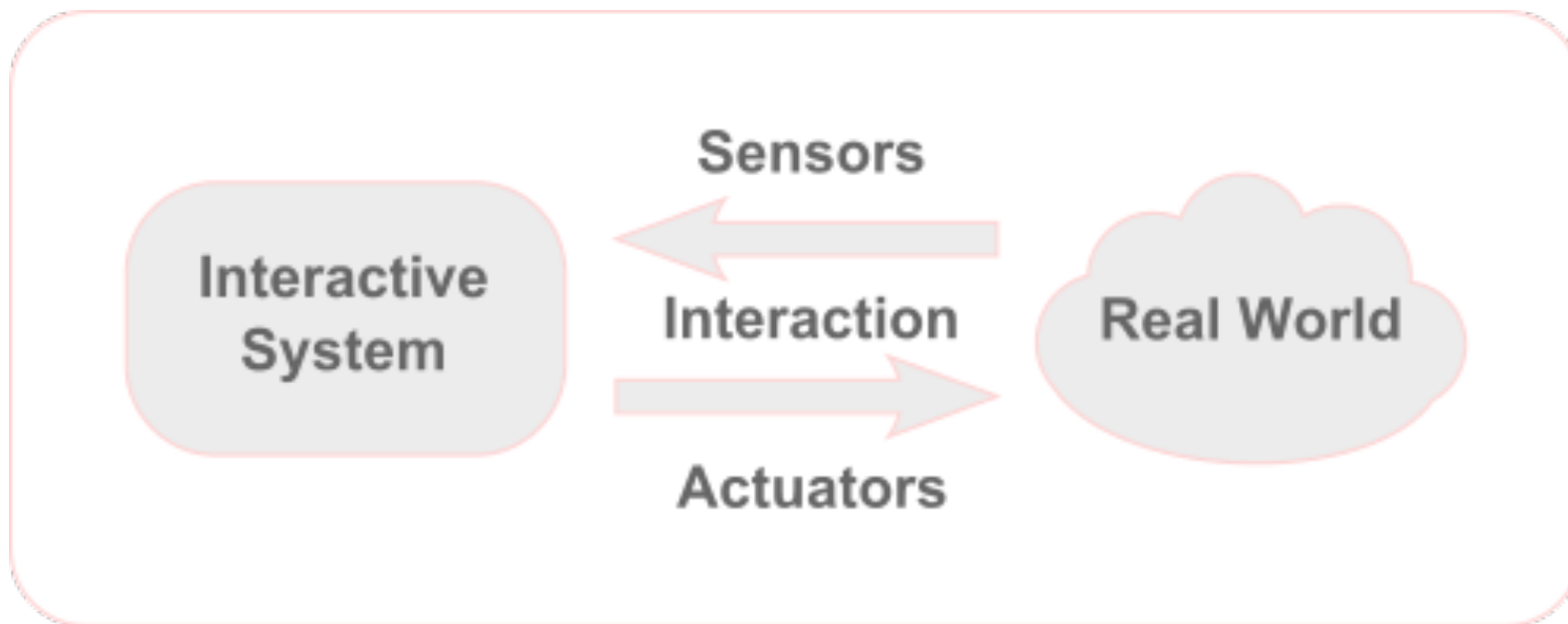
- е изграждането на интерактивни физически системи, използвайки софтуер и хардуер, които могат да усещат и реагират на реалния свят
- макар че тази дефиниция обхваща приложения като например една интелигентна система за контрол на трафика или една система за автоматизиране на процесите в един завод не се използва за да ги опише

Физикъл компютинг

- с други думи, физикъл компютинг представлява креативната рамка, която подпомага разбирането на отношението на хората към цифровия свят
- на практика, означава по-често проекти ръчна разработка, или DIY (Do It Yourself), които използват сензори и микроконтролери за да предадат аналоговия вход към софтуерни системи, и/или за да контролират електро-механични устройства като мотори, серводвигатели, осветления и др.
- http://en.wikipedia.org/wiki/Physical_computing

ФИЗИКЪЛ КОМПЮТИНГ

- схематично таво се изразява като:



ФИЗИКЪЛ КОМПЮТИНГ

- е подход за проектиране на взаимодействие на човека с компютъра, който преразглежда проблема как хората да изразят себе си физически
- базираната на хардуера на компютъра, конфигурация на взаимодействието не се възприема за дадена
- именно, че има клавиатура, екран, тонколони и мишка или тракпад или сензорен екран
- човешкото тяло и неговите възможности представляват отправна точка, и се опитва да се разработи интерфейс, софтуер и хардуер, който да може да усети и да реагира на това, което, човека, може

Физикъл компютинг срещу роботика

- като цяло изискванията към хардуера, използван във физикъл компютинг са подобни на тези, за използвания хардуер в роботиката
- концепцията обаче е доста по-различна

Физикъл компютинг срещу роботика

- при изграждане на роботи, фокуса е върху създаване на устройства, които са самостоятелни, способни да навигират себе си в света
- физикъл компютинг системите обаче фокусират върху взаимодействието с човека

Физикъл компютинг срещу роботика

- вместо автоматизация фокусираме върху използването на цифрови технологии, за да разширим човешките способности
- създаването на системи, които се управляват от намерения, решения и действия на човека

Физикъл компютинг срещу роботика

- ключовата разлика се състои в това че:
- роботиката цели да се съсредоточи върху механиката, системите за управление и наблюдение на работа, докато
- физикъл компютинг, се концентрира повече върху интерфейса, т.е. хардуера и софтуера, които са необходими на човека за да насочва / управлява машината

Инсцениране на поведения

- проектиране срещу “learn by doing”
- професионалния подход срещу занаята
- има ли друга междинна форма?

Инсцениране на поведения

- човека планира и мисли за това, което предстои задължително има някакъв проект преди изграждането на продукта или извършването на действието
- компютрите променят процеса на проектирането. По-лесно е да копирате и промените, да имитирате и да се адаптирате докато се развива от "полу-работещ код" през следващата итерация на системата, работещ код

Инсцениране на поведения

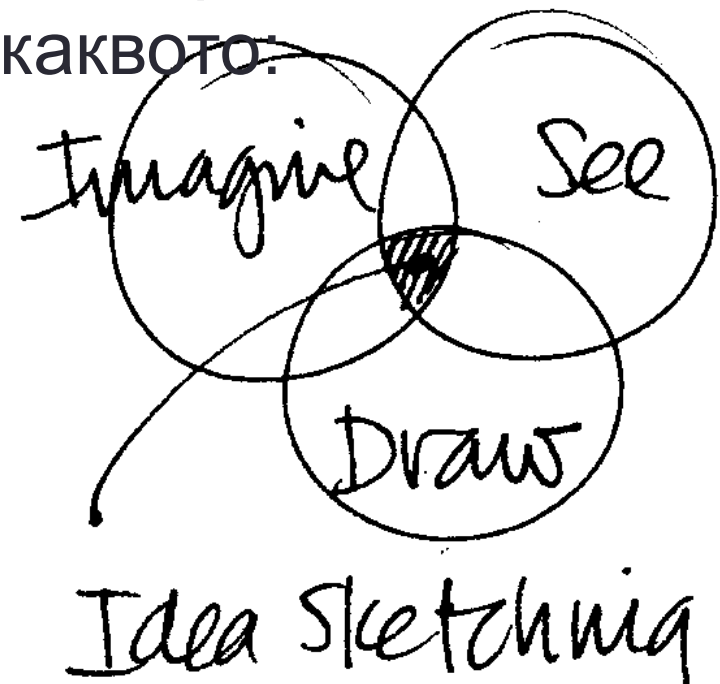
- прякото ангажиране с произвеждането на незабавни резултати, е това, което прави занаятчийската традиция. Не се инвестира време в планиране или абстрактно анализиране
- професионалната практика обаче задължително трябва да инвестираме в проектиране
- инсценирането може да бъде първата стъпка при новия подход на проектиране

Инсцениране на поведения

- проектирането на взаимодействие фокусира върху човека
- проект за човешкото поведение
- взаимодействието с технологията или с други лица чрез технологията ни изправя пред компютри, което правя самото проектиране на взаимодействието предизвикателство

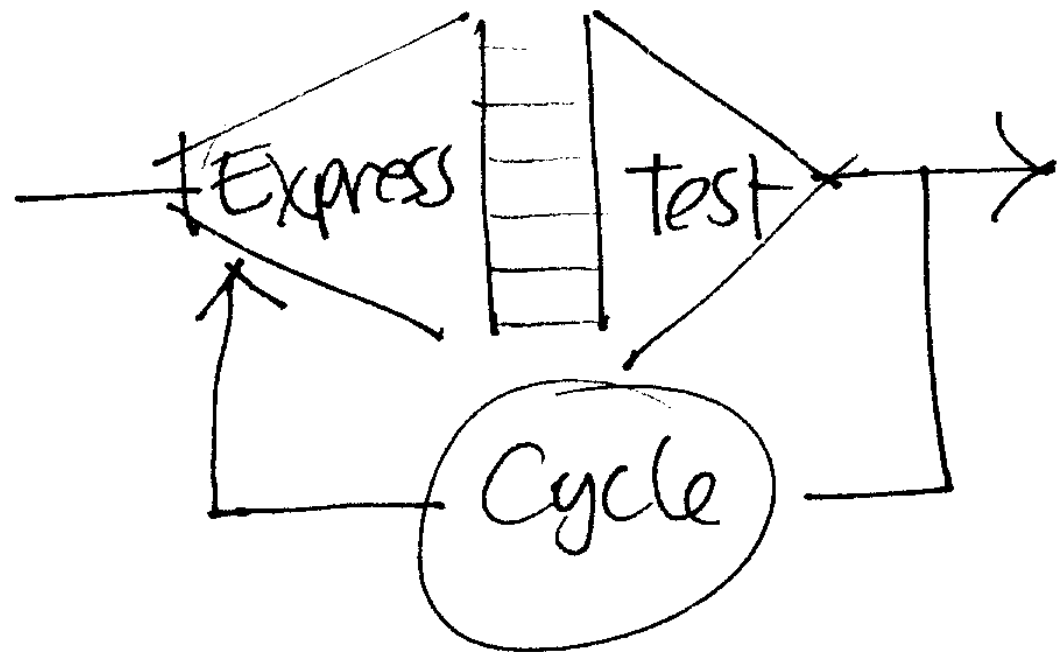
Инсцениране на поведения

- това, което виждаме се влияе от това, което ние си представяме, но същевременно това, което ние си представяме, зависи от това, което виждаме
- инсценирането на поведения трябва същевременно да включва каквото:
 - виждаме
 - си представяме и
 - оформяме



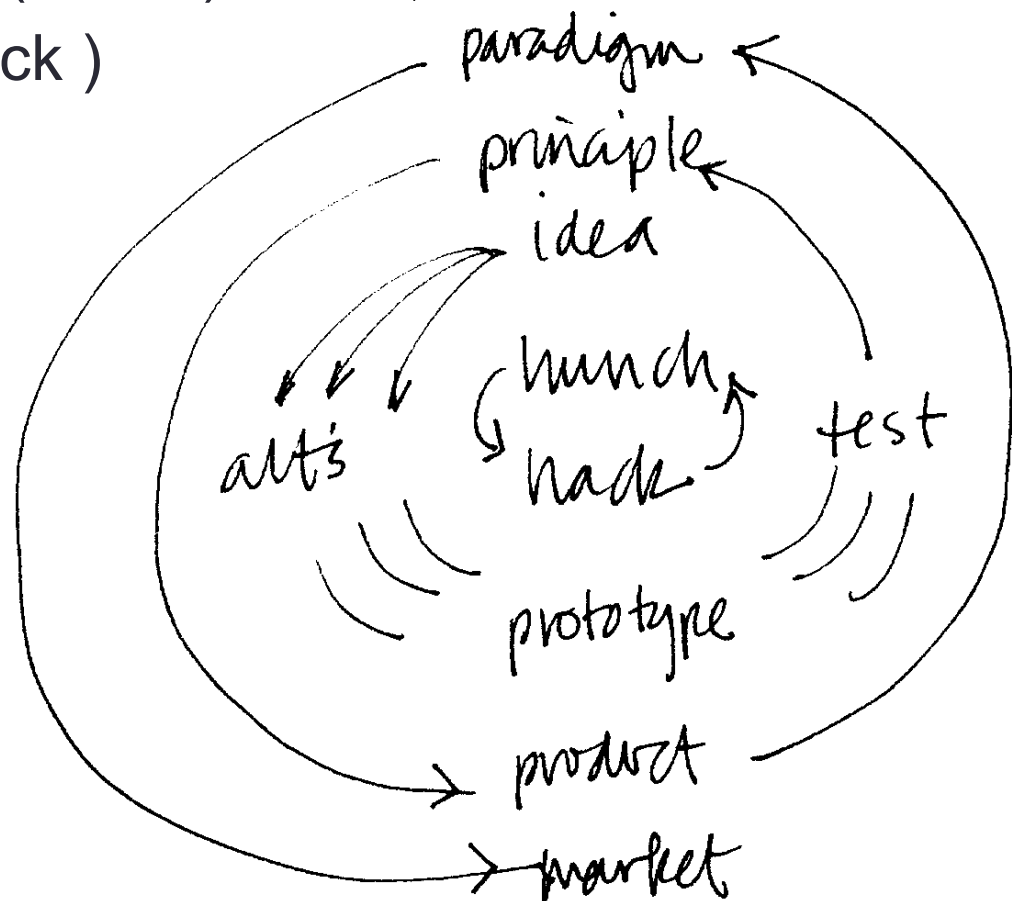
Инсценириране на поведения

- тогава, проектирането може да представлява един цикъл от тестване на предложени изпбращения



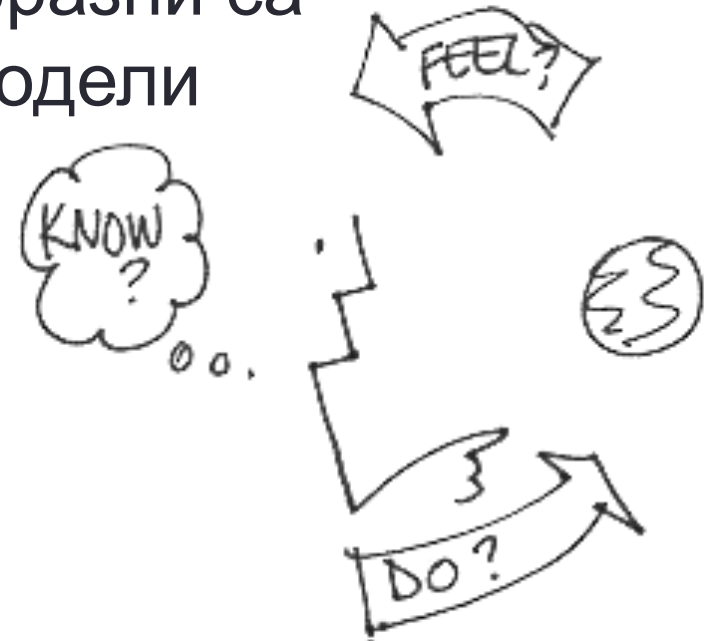
Инсценириране на поведения

- и това да ни е процедурата дори за узряване на продуктите за пазара
 - от една дреболийка (hunch)
 - по предчувствие (hunch)
 - ...
 - до пазара (market)



Взаимодействие

- най-простият уред изисква, усещане и знание
- включването на лампата, означава че трябва да видим (усетим) светлината
- колкото по-голямото разстояние от входа (превключвател) до изхода (светлина), толкова по-трудни и разнообразни са възможните концептуални модели

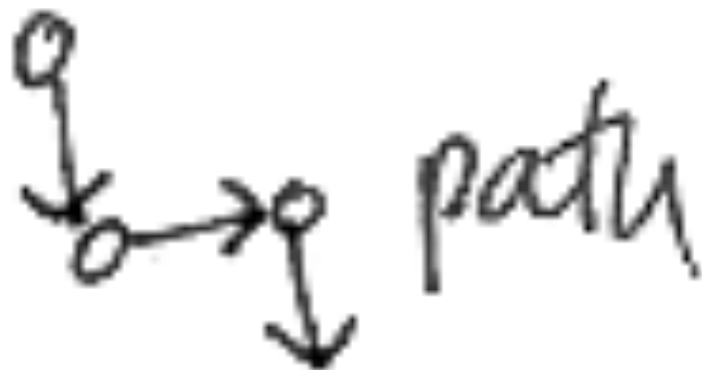


Взаимодействие

- при работа с изчислителни машини, можем да проектираме продукти, които да отговарят на почти всяко поведение
- отговорът за човешкия вход може да бъде забавен или повтарящ се (чрез съпоставяне)
- системите могат да променят начина, по който реагират (режими на работа)
- заради мрежите, идеята за самостоятелен продукт е остаряла
- ефектът от моите действия, може да бъде местен или дистанционен

Взаимодействие

- най-лесното взаимодействие изисква да се знае само една стъпка във всеки момент (path knowledge)
 - например, процедури при спешни случаи
 - стъпка по стъпка инструкции
- по-развитите изискват потребителя да използва знания на карта (map knowledge)



Проектиране на взаимодействие

- успешното проектиране на взаимодействие включва балансиране на различни аспекти, с помощта на различни методи за действие или представяне
- като цяло това не представлява препоръка за разделяне на етапите от процеса на проектирането, но като рамка за проверка, за да се види, че всички аспекти са били адресирани

Проектиране на взаимодействие

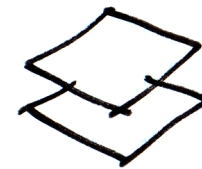
- в горната част са по-абстрактни прегледи, а в долната са подробности
- от ляво на дясно колоните може да се нарекат мотивация, значения, режими и преобразувания,
- а самият процес от ляво на дясно може да се дефинира като преминаване от наблюдение, изобретение, инженеринг до оформяне на външния вид



IDEA



METAPHOR



MODEL



DISPLAY



ERROR



SCENARIO



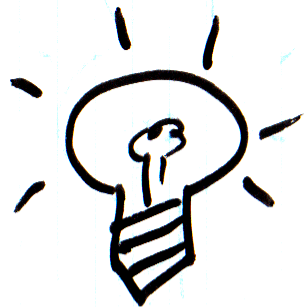
TASK



CONTROL

Проектиране на взаимодействие

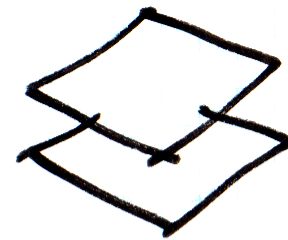
- Резултатът от проектиране на дадено взаимодействието се състои от дисплеи и контроли, както и от поведениията, които ги свързват (mapping)



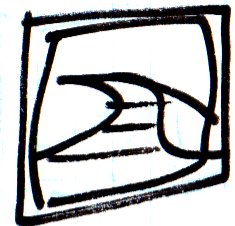
IDEA



METAPHOR



MODEL



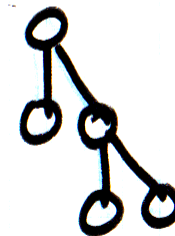
DISPLAY



ERROR



SCENARIO



TASK



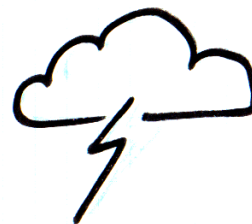
CONTROL

Проектиране на взаимодействие

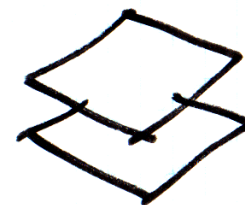
- С цел създаване на условия за съгласуваното прилагане трябва да има и двата компонента:
 - стъпка по стъпка анализ на взаимодействията, и
 - концептуалния модел, който организира поведението в режими, както за създателите, така и за потребителите



IDEA



METAPHOR



MODEL



DISPLAY



ERROR



SCENARIO



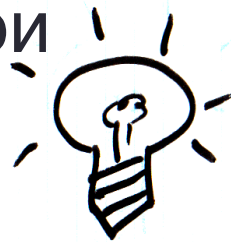
TASK



CONTROL

Проектиране на взаимодействие

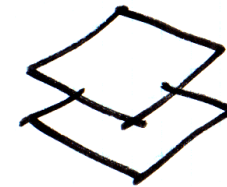
- така проектирането на взаимодействие включва не само един завладяващ сценарий и обединяващата метафора, но също така разглеждане на различни сценарии и по-широко проучване на алтернативни и смесени метафори



IDEA



METAPHOR



MODEL



DISPLAY



ERROR



SCENARIO



TASK



CONTROL

Проектиране на взаимодействие

- успешни примери на така дефинирано проектирането на взаимодействие са преобразували нашите разбирания за изчислителните машини като:
 - МОЗЪК
 - инструмент и т.н.



Прототип

- Прототип означава първоначален вид на нещо
- Думата най-често се използва за обозначаване на работни модели, създадени за да демонстрират определен нов продукт
- Прототипи се използват в множество индустрии, напр. машиностроене, разработка на софтуер и т.н.
- Успешната разработка на прототип позволява събиране на ранна информация за експлоатацията, ползваемостта и надеждността на новия продукт

Прототип

- Една от няколко наложени в практиката терминологии, разделя прототипите на алфа (alpha, α) и бета (beta, β) версии, следвани от официалното пускане на продукта
- В индустрии, в които разработката на прототипи не е свързана с много големи разходи, като например разработката на софтуер, често се използва техника, наречена бързо прототипиране (rapid prototyping)
- <http://bg.wikipedia.org/wiki/Прототип>

Предимства и недостатъци

- Предимства :

1. Намалява времето за разработка
2. Намалява разходите за развойна дейност
3. Изисква участието на потребителя
4. Разработчиците получават количествено обратна връзка от потребителите
5. Улеснява системата за изпълнение, тъй като потребителите знаят какво да очакват
6. Положителен ефект в по-висока удовлетвореност на потребителите
7. Излага разработчиците на потенциални подобрения бъдещата система

Предимства и недостатъци

- Недостатъци:

1. Може да доведе до недостатъчен анализ
2. Потребителите очакват изпълнението на крайната система да бъде същата като прототипа
3. Разработчиците могат да станат твърде привързани към своите прототипи
4. Може да предизвика системи да остават недовършени и / или да се изпълняват, преди те да са готови
5. Понякога води до непълна документация
6. Ако се използват сложни системи за създаване на прототипи ползата от създаването им може да се загуби

Прототайпинг

- като процес включва следните стъпки:

1. Идентифициране на основните изисквания

Определя основните изисквания, включително на желаната информация на входа и на изхода. Детайли, като сигурност, обикновено могат да бъдат пренебрегнати

2. Разработване на първоначален прототип,

Първоначалният прототип е разработен, който включва само потребителски интерфейси.

Използва се и термина хоризонтален прототип, като общ термин за прототип на потребителския интерфейс. Той предоставя широк поглед върху цялата система или подсистема, като се фокусира върху взаимодействието на потребителите и не на функционалността на системата на ниско ниво, като например достъп до база данни

3. Преглед

4. Преразглеждане и подобряване на прототипа

Прототайпинг

- като процес включва следните стъпки:

1. Идентифициране на основните изисквания
2. Разработване на първоначален прототип,
3. Преглед

Клиентите, включително крайните потребители, проучват прототипа и предоставят обратна информация за допълнения и промени

4. Преразглеждане и подобряване на прототипа

Чрез използването на обратната връзка, както и на спецификациите прототипът може да бъде подобрен. Преговори за това, което е в рамките на обхвата на договора / продукт може да бъдат необходими. При въвеждане на промени стъпките # 3 и # 4 може да се наложи да се повторят

Прототайпинг

- като процес включва следните стъпки:

1. Идентифициране на основните изисквания
2. Разработване на първоначален прототип,
3. Преглед

Клиентите, включително крайните потребители, проучват прототипа и предоставят обратна информация за допълнения и промени

4. Преразглеждане и подобряване на прототипа

Чрез използването на обратната връзка, както и на спецификациите прототипът може да бъде подобрен. Преговори за това, което е в рамките на обхвата на договора / продукт може да бъдат необходими. При въвеждане на промени стъпките # 3 и # 4 може да се наложи да се повторят

Ардуино прототайпинг

- ардуино представлява пакетна платформа, която се състои от стандартизиран софтуер и хардуер
- това улеснява процеса за обучение, чрез правене
- learn by doing от своя страна отваря вратите на широка публика

Ардуино прототайпинг

- независимо от опрастяванията на модела на работа, добавъчната стойност идва от това че каквото и да разработиме е приложимо
- това което се прави като проект в ардуино, може да се превърне в реален продукт, или част от реален продукт

Ардуино прототайпинг щит

- Идеята на ардуино прототайпинг щит е да улесни изработването на вериги по поръчка
- можем да прикачим със спойка частите към щита, за да създадем проекта си, или да го използваме с малък breadboard за още по-бързо тестване на примерни вериги, без да се налага спойка
- щита има допълнителни връзки за всички Arduino I/O пина, и място за монтиране на интегрални схеми

От ардуино прототип към продукт

- Стъпките са:

1. изграждане на прототип използвайки arduino и breadboard [няколко дни]
2. нарисуване и схематично проектиране на печатна платка с CAD програма [1 ден]
3. поръчка на protoboards [2-4 седмици]
4. тест и адаптиране на protoboards [1 час до дни]
5. събиране на оферти за производство [1 седмица]
6. избор на производител и прашане на поръчка [няколко седмици]

От ардуино прототип към продукт

- с други думи говориме за експерименти и тестове, защото самото правене на прототипи е експериментиране и тестване
- фокусът за пореден път е креативността, при наличие на елементарни условия