

Спецификација захтева

Развој софтвера (4)

проф. др Милан Гњатовић

Факултет за компјутерске науке
Универзитет Џон Хезбит

22.03.2016.

Садржај

- 1 Увод
- 2 Особине ефективног захтева
- 3 Комплетност спецификације
- 4 Пример: Како се специфира дијалошки систем?
- 5 „Чаробњак из Оза“
- 6 Завршне напомене

Дефиниција и типови захтева

- Захтев је израз жељеног понашања система.
- Типови захтева:
 - Функционални захтев — **Шта** систем треба да ради?
 - Захтев у погледу квалитета — **Колико добро** то треба да ради?
 - Нпр., поузданост, перформансе, сигурност, робустност, лакоћа коришћења, трошкови одржавања, итд.
 - Пројектна и процесна ограничења — Да ли постоје **ограничења** у вези са ресурсима или дизајном?
 - Пројектна ограничења: нпр., унапред изабране платформе и компоненте, итд.
 - Процесна ограничења: нпр., клијент инсистира на примени агилне методе, итд.

Чему служе захтеви?

- Захтев не одређује како ће систем бити имплементиран.
 - Ове одлуке се доносе у фази дизајна, на основу захтева.
- Захтеви помажу да се успостави међусобно разумевање између клијента и развојног тима о томе шта треба да се постигне софтвером.
- Захтеви се постижу споразумом са клијентима — и о њима се преговара.
 - Клијенти не могу увек јасно да формулишу своје потребе, а понекад их нису ни свесни.
 - Развојни тим понекад не може да схвати потребе клијената.

Особине ефективног захтева

- Комплетност,
- коректност,
- концизност,
- изводљивост,
- неопходност,
- недвосмисленост,
- проверљивост,
- конзистентност,
- следљивост.

Комплетност

- Захтев је комплетан кад садржи довољно података на основу којих се могу донети одлуке у каснијим фазама развоја.
- Пример:
 - Претпоставимо да је у спецификацији апликације за електронско пословање дефинисана нека функционалност, али је (неоправдано) изостављено да велики број корисника може истовремено да користи ту функционалност.
 - Као резултат, софтвер имплементира дату функционалност, али су перформансе у случајевима већег броја корисника неприхватљиве.
 - Пошто су перформансе система нераскидиво везане за његову архитектуру — да би се исправио овај недостатак, потребно је мењати архитектуру система.
 - Ова измена захтева повратак на почетак развојног циклуса.

Коректност

- Захтев је коректан када не садржи грешке.
- Грешке могу бити: типографске грешке, нетачне претпоставке, неразумевање потреба клијента, итд.
- Пример:
 - Захтеви не смеју да буду базирани на претпоставкама дизајнера.
 - У случају да претпоставке нису биле коректне, грешка ће се открити тек када клијент проба да користи софтвер.

Концизност

- Захтев је концизан када садржи само неопходне информације, и када је изражен на једноставан начин.
- Узроци неконцизних захтева су:
 - преопширне реченице (нпр., више захтева изражено у једној реченици),
 - комплексна граматика,
 - редундантни захтеви,
 - додавање непотребних образложења, примера, дизајнерских аспеката, итд.
- Пример — проблеми када дизајн постане део спецификације:
 - Постоји ризик да изабрано решење није најбоље.
 - Сугерише се тестирање система у односу на дизајн, а не спецификацију (што је погрешно).
 - Један од индикатора ове грешке је када се технологија референцира у спецификацији.

Изводљивост

- Захтев је изводљив уколико за њега постоје бар један дизајн и једна имплементација који задовољавају потребе клијента.
- Захтев није изводљив када се:
 - не може технички реализовати,
 - не уклапа у предвиђени распоред,
 - не уклапа у предвиђени буџет, итд.

Недвосмисленост

- Захтев је недвосмислен када се може интерпретирати на само један начин.
- Пример — Услов из конкурса за конструкцију летеће машине (1907.)¹:
 - „Треба да буде довољно једноставно за конструкцију и управљање тако да интелигентан човек може да постане вешт у управљању у разумном времену.“
- Како интерпретирати следеће захтеве?
 - „довољно једноставан“,
 - „интелигентан“,
 - „вешт“,
 - „разумно време“.

¹Advertisement and Specification for a Heavier-Than-Air Flying Machine, U.S. Army Signal Corps Specification No. 486, 1907.

Неопходност

- Захтев је неопходан када се намена софтвера не може остварити без задовољавања тог захтева.
- Пример:
 - Развојни тим може (понекад неоправдано) да инсистира да клијент на почетку дефинише све захтеве.
 - Реакција клијента је да заштити свој интерес и проба да се сети *свих могућих* захтева, иако је за неке мало вероватно да ће му икад бити потребни.
 - У том случају ће резултујућа спецификација бити оптерећена сувишним захтевима.

Проверљивост

- Захтев је проверљив када може да се тестира да ли софтвер задовољава дати захтев.
- Пример лоше дефинисаног захтева:
 - „Софтвер ће бити лак за коришћење од стране искусних службеника и организован на начин који минимизује грешке.“
 - Како се ово тестира?
- Уместо тога:
 - „Искусни службеници ће бити у стању да користе све функционалности система након двочасовног тренинга. Након тренинга, просечни број грешака које направи искусни службеник неће бити већи од две грешке по дану.“

Конзистентност и следљивост

- Захтев је конзистентан уколико није противречан неком другом постојећем захтеву.
- Захтев је следљив уколико је једнозначно и доследно означен.
 - Везе са повезаним захтевима морају да буду дефинисане и документоване.

Комплетност спецификације

- Ниједна спецификација није сасвим комплетна ...
 - ... нити је потребно да буде.
 - Ионако немате довољно времена ни ресурса.
- Обратите пажњу на детаље у вези са карактеристикама и функционалностима које су:
 - ризичне,
 - нове (немате искуства са њима),
 - комплексне.

Ниво детаља

- Исправно одређени ниво детаља је баланс између ризика у развоју и инвестиције која се улаже у дефинисање захтева.



Дијалошки системи

- Дијалошки системи² (конверзациони агенти):
 - Дијалошки системи су рачунарски системи са којима корисник комуницира користећи природни говор.
 - Основна намена ових система је да посредују између корисника и екстерног рачунарског система (нпр., база података, систем за резервацију карата, аутоматско преусмеравање позива, итд.).

²Више о дијалошким системима на наредим предавањима.

Шта су захтеви за дијалошки систем?

- Захтеви за дијалошки систем су многобројни, и укључују:
 - речник израза и фраза које корисници могу да употребе,
 - граматику по чијим правилима корисници могу да формирају своје дијалошке чинове,
 - могуће проблеме у комуникацији,
 - дијалошке стратегије за превазилажење проблема,
 - моделовање дијалога, итд.
- Како се дефинишу ови захтеви?

Приступ 1. Интуиција

- Захтеви се базирају на интуицији клијента и програмера.
- Овај приступ није прихватљив.
 - Рефлектује теоријска интересовања и предрасуде појединца.
 - Не може се претпоставити да ће корисници моћи и желети да комуницирају на предвиђени начин.

Приступ 2. Анализа конверзације између људи

- Захтеви се формирају на основу анализе снимака конверзације између људи.
- Иако бољи од претходног, овај приступ није сасвим задовољавајући.
 - Конверзација између људи се разликује од конверзације између човека и машине.
 - Не важе иста правила.

Приступ 3. Анализа конверзације између човека и машине

- Захтеви се формирају на основу анализе снимака конверзације између човека и датог система.
- Ово делује прихватљиво.
- Али како доћи до снимака конверзације између човека и система, када систем још није имплементиран?
- Слична дилема се јавља и код валидације захтева.
- Чак и када имамо дефинисане захтеве, да ли је потребно да имплементирамо цео систем да бисмо могли да оценимо њихову валидност?
- Решење пружа симулациона техника позната под називом „Чаробњак из Оза“.

Етимологија

- Да ли се сећате приче о чаробњаку из Оза?
- Чаробњак је у ствари само умешни илузиониста који се крије иза паравана.

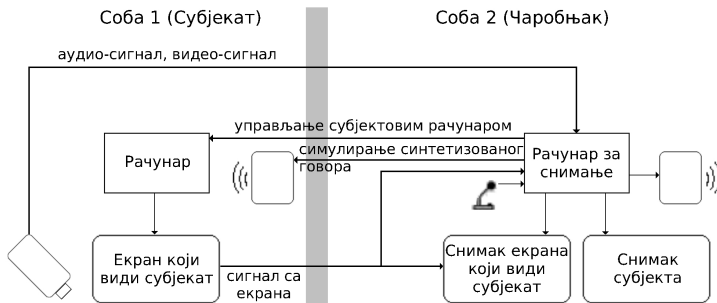


3

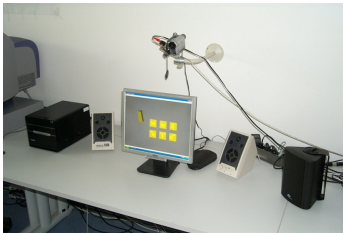
³Сцена из филма „Чаробњак из Оза“ (1939).

Основна идеја

- Симулира се интеракција између човека и система:
 - Субјектима се пружа илузија да причају са системом.
 - Међутим, функционалност система је симулирана — човек (у улози чаробњака), који је скривен од субјекта, игра улогу система.
- Пример — Упрошћена шема лабораторије за симулацију:



Стварно окружење



Предуслови за успешну симулацију

- Симулација система мора да буде могућа, имајући у виду људска ограничења.
 - Чаробњак мора да буде у стању да реалистично имитира систем.
 - Нпр., време одзива симулираног система мора да буде реалистично.
- Понашање симулираног система мора да буде „специфицирано“.
 - Нпр., уколико се очекују грешке у препознавању гласа (што је иначе неизбежно у реалним условима), потребно је специфирати које типове грешака чаробњак треба да симулира, и којом учестаношћу.
- Симулација мора да буде убедљива.
 - Илузија коју имају субјекти не сме бити разоткривена до краја симулације.
 - Иначе се неће понашати као да комуницирају са системом.

Литература

- Leishman, T.R., Cook, D.A., Requirements Risks Can Drown Software Projects, Computer (November 2001).
- Fraser, N.M., Gilbert, G.N., Simulating speech systems, Computer Speech & Language 5(1), pp. 81–99 (1991).

Хвала на пажњи

- Питања су добродошла.