ДОКЛАД

ПО СОФТУЕРНО ИНЖЕНЕРСТВО

1. Прототипиране на потребителски интерфейс
2. Методологий за разработка на софтуер – Waterfall и Agile
3. Scrum. Scrum артефакти, екипи, роли и събития
4. Прототипиране на потребителски интерфейс
5. **Прототипирането на потребителски интерфейс** е процес, при който се създава модел или прототип на интерфейса на едно приложение или система. Този прототип имитира реалното приложение и позволява на дизайнерите и разработчиците да тестват и подобряват потребителското изживяване.
6. В процеса на прототипиране могат да се използват различни инструменти и техники. Например, можеш да създадеш визуални елементи на интерфейса чрез дизайнерски софтуер и да ги свържеш с интерактивни функционалности. Това позволява на потребителите да взаимодействат с прототипа и да предоставят обратна връзка, която може да бъде използвана за подобрения.
7. Прототипирането на потребителски интерфейс е полезно, защото позволява на екипа да тества и изследва различни варианти на дизайна, преди да го разработят напълно. Това спомага за откриване на проблеми или несъответствия в интерфейса и за оптимизиране на потребителското изживяване. Освен това, прототипите могат да бъдат споделени с потребителите за получаване на обратна връзка и валидация на концепцията преди финалната разработка.
8. Методологий за разработка на софтуер – Waterfall и Agile
9. **Waterfall**

* Методологията на водопада е широко използван метод за управление на проекти с линеен подход. Във Waterfall всеки етап от работния процес трябва да бъде завършен, преди да преминете към следващата стъпка. Въпреки че има различни видове методологии за управление на проекти, Waterfall е много подходящ за проекти, при които целите са ясно очертани от самото начало. Тази статия обхваща как работи Waterfall, за какви проекти методологията е най-подходяща и как се сравнява с Agile, друг метод за управление на проекти.

**Предимства**

* Едно от предимствата на Waterfall е, че има фиксиран график и бюджет, тъй като целите на проекта са конкретни и очертани от самото начало.
* След установяване на целта на проекта, методологията на водопада не включва честа обратна връзка или сътрудничество от страна на клиента, освен установените етапи или резултати за всяка фаза.
* Това улеснява ръководителите на проекти да планират и комуникират със заинтересовани страни или бизнес партньори.
* Въпреки това, въпреки че това може да помогне при планирането, то е и практично само когато клиентът има ясна и фиксирана крайна цел и не е необходимо да участва в процеса на разработване на проекта.

**Недостатъци**

* Един от недостатъците на тази методология е, че адресирането на неочаквани проблеми може да бъде трудно и навременно.
* Докато се преминава от фаза на фаза, фазите могат да се информират една друга“, например предизвикателствата във фазата на внедряване може да показват, че е имал лош дизайн. Проблемът е, че може никога да не се разбере това, докато не се внедри или извърши проверка и валидиране“.
* В тези ситуации твърдостта на пълното завършване на една фаза преди преминаване към следващата може да удължи сроковете на проекта. Най-добрият начин да предотвратите това, когато се използвате Waterfall, е да се получи възможно най-много информация, за да се избегне риска от необходимост да се върне към фаза.

1. **Agile**

* етодологията е рамка за управление на проекти, която разделя проектите на няколко динамични фази, известни като спринтове.

Agile рамката е [итеративна методология](https://asana.com/resources/iterative-process) . След всеки спринт екипите размишляват и поглеждат назад, за да видят дали има нещо, което може да се подобри, за да могат да коригират стратегията си за следващия спринт.

**Предимства**

* Повишено качество на софтуера – тъй като при прилагане на Agile методология се работи на кратки интервали, в края на които се доставя софтуер, грешките са много по-лесни за засичане и коригиране.
* Намалени разходи по разработката – тъй като проектите стават по предвидими, а изпълнението на инкременти позволява по-бързото засичане на потенциални проблеми и отстраняването им, това значително намалява разходите, които са неотменна част от софтуерната разработка.
* По-бързо достигане до пазара – итеративният подход при прилагането на Agile методология значително намалява времето за достигане до потребителите. Те получават функционалностите, които искат, в кратки срокове, а ти и екипът ти се движите пред конкуренцията.

**Недостатъци**

* Непрекъснатата комуникация между различни екипи и с различни заинтересовани лица може да натовари разработчиците и да затрудни управлението на времето.
* Не на последно място, ограничението във времето и обхватът на проектите, управлявани чрез Agile практики, може непрекъснато да се променят. От една страна, това е проблем, когато проектът трябва да бъде доставен на фиксирана дата. От друга – може непрекъснато да изникват нови и нови функционалности и изисквания, което да затрудни екипите в приоритизирането на задачи.
* Въпреки тези предизвикателства, това все пак остава предпочитана методология на работа в софтуерната индустрия. Ако и ти искаш да навлезеш в сферата и управлението на софтуерни проекти, да разбереш какво е да работиш по Agile методология и какви са разликите с по-традиционните подходи, не пропускай предстоящия курс.

1. Scrum. Scrum артефакти, екипи, роли и събития

* Scrum e най-популярната рамка за работа по Agile методологията. Причината Scrum да е толкова популярна рамка в IT екипите е фактът, че помага за **по-лесното и ефективно осъществяване** на много големи проекти, разделяйки проекта на итеративни фази, наречени спринтове (sprints). Това са времево-регламентирани (time-boxed) събития, в които екипът изпълнява определен набор от задачи, които са част от цялостния проект. Scrum рамката има три основни аспекта:
* **Роли** – според рамката, екипът се разделя на три основни роли – собственик на продукт (product owner), Scrum master и разработчиците (development team). Именно тях ще разгледаме по-подробно в следващите редове.
* **Събития** – събитията в Scrum екипите включват планиране на спринт, самият спринт, ежедневен Scrum, или ежедневни срещи, познати като daily, ревю и ретроспекция на спринта. Събитията служат за подобряване на процесите по работа и ефективното осъществяване на проекта.
* **Артефакти** – когато работиш по Agile методология като Scrum, разполагаш с три артефакта – Product Backlog, който съдържа всичко, което трябва да се свърши по даден проект/продукт, и е източникът на работа за Scrum екипа; Sprint Backlog, който съдържа етапите от текущата работа, задачите от Product Backlog-а, които се вършат сега; и Increment (или инкремент – конкретният етап, на който се намира екипа от цялостната работа по продукта). Артефактите служат за приоритизиране на задачите и доставянето на резултати в уговорени срокове.

1. Продуктов собственик (Product Owner)

**Product Owner-ът** отговаря за Product Backlog-а на продукта. Той осигурява ясната формулировка на продуктовата цел, последователността на елементите в Product Backlog-а и ясното им комуникиране. Неговата отговорност е да максимизира стойността на продукта. Един човек въплъщава ролята, но има възможност да делегира задълженията си. Ако някой иска да се направят промени в продуктовия Backlog, трябва да убеди Product Owner-а, че ще са от полза.

1. Scrum Master

**Scrum Master-ът** е роля, която също се пада на един човек в екипа. Неговата задача е да осигури ефективността на екипа, да помага на колегите около себе си (не само в екипа, но и в организацията си) да разбират приложението, а и теорията зад рамката, както и да подобряват практиките си в рамките на Scrum.

1. Разработчици (Development Team)

**Разработчиците** са хората, които всъщност вършат работата по проекта. От тях зависи съдържанието на плана за спринта (Sprint Backlog). Те ежедневно адаптират този план, за да постигнат целта, залегнала в спринта. Разработчиците допринасят с индивидуалните си умения, но отговорността по реализацията на целите е колективна за екипа. Заедно те формират мултидисциплинарен екип, който ежеседмично работи по елементи от Product Backlog-а, за да могат в края на спринта да реализират инкремента.

Източници

1. Прототипиране на потребителски интерфейс - <https://3dbgprint.com/burzo-suzdavane-na-prototipi-s-3d-printirane/>
2. Методологий за разработка на софтуер – Waterfall и Agile - <https://www.forbes.com/advisor/business/what-is-waterfall-methodology/>

<https://softuni.bg/blog/agile-methodology-advantages-disadvantages>

<https://asana.com/resources/agile-methodology>

1. Scrum. Scrum артефакти, екипи, роли и събития –

<https://softuni.bg/blog/scrum-roles-and-meaning>

Изготвил: Валентин Кьосев12 ,,а‘‘ клас