



МИНИСТЕРСТВО НА ТРУДА И СОЦИАЛНАТА ПОЛИТИКА
**ЦЕНТЪР ЗА РАЗВИТИЕ НА ЧОВЕШКИТЕ РЕСУРСИ
И РЕГИОНАЛНИ ИНИЦИАТИВИ**

София 1849, кв. Кремиковци, тел./факс: +359 2 994 70 18, +359 882 82 66 83, +359 876 24 82 04
www.chrdri.net

СОФТУЕРНИЯТ ПРОЦЕС

Дефиниция и основни елементи

В софтуерното производство терминът софтуерен процес се отнася до метода за разработване на софтуер.

Под софтуерен процес се разбира множество от дейности заедно със съществуващите ограничения между тях, които водят до производството на висококачествен софтуер на ниска цена. Той се занимава технически и управленски задачи и определя метода при разработването на софтуер. Софтуерният процес задава абстрактно множество от дейности, които трябва да се изпълняват, за да се реализират потребителските изисквания в крайния продукт. Използвайки един и същ тип софтуерен процес (метод за разработване) могат да се разработват различни софтуерни проекти, като в рамките на всеки проект могат да се получат различни видове продукти (фиг.2.1).

Фиг.2.1 Софтуерния процес

Неминуемо възниква следният въпрос: защо след като софтуерният процес определя (предписва) отделните дейности при разработването на софтуер, е трудно тези дейности да се спазват при конкретен проект? Причините за това са следните:

1. Софтуерният процес определя дейностите на абстрактно ниво, защото те трябва да са приложими за голям брой, различни по своето естество, проекти. Следователно прилагането им в конкретен проект не е толкова директно. Може да се каже, че процесът определя списък от дейности, чиято реализация изисква значителни усилия. Пример: пътуване - абстрактно и пътуване до морето.
2. Софтуерният процес определя дейностите, без да отчита спецификата на конкретния проект. Тези дейности се уточняват в плана на проекта, където се отчитат какви точно дейности да се извършват за този конкретен проект.
3. Процесът управлява проекта като регламентира дейностите, които трябва да се извършат и техния ред.

Може да се каже, че софтуерният проект е състояние на софтуерния процес (сравни с схема и състояние при модели на данните).

Извод: обектите на софтуерното производство са: процеси, проекти, продукти. Всеки един от тях следва собствен метод, т.е. изисква различен процес.

Софтуерният процес се състои от три основни вида процеси:

1. Процес за разработване - негов обект е продукта (софтуер, ръководство за потребителя, спецификация на изискванията и др. подобни продукти). Целта на този процес е производството на продукт, който да удовлетворява потребителя.
2. Процес за управление на проекта - негов обект е проекта. Целта на този процес управление на всички дейности свързани с проекта за да се постигне набелязаната цел.
3. Процес за управление конфигурирането на софтуера - управлява неизбежните промени или преработки, които могат да настъпят в един проект, така че да не се наруши целостта на продукта поради преработването му.

Ясно е, че последните два процеса зависят от процеса за разработване. Затова на него като основен процес се обръща особено внимание и повечето модели се предлагат именно за

него.

Тези три процеса наблягат върху проектите и продуктите. Те могат да се разглеждат като процеси, свързани с производството на продукт, понеже тяхната основна цел е да се произведе необходимия продукт.

Отделните процеси се различават не само по своите дейности, но и по хората, които ги изпълняват. Дейностите по разработването се изпълняват от програмисти, проектантите, тестващ персонал и др. Дейностите по управлението на проекта се изпълняват от мениджър на проекта и т.н. Ще разглеждаме основно процесите, свързани с производството на продукт. Терминът софтуерен процес ще използваме в по-тесен смисъл - като общо наименование на процесите свързани с производството (разработване) на продукт.

Характеристики на софтуерния процес

Основните цели на софтуерния процес съвпадат с тези на софтуерното производство - те са оптималност и мащабируемост. Оптималност означава, че процесът трябва да може да произвежда софтуер с високо качество на ниска цена, а мащабируемост означава, че процесът е приложим за големи софтуерни проекти.

За да се постигнат тези цели процесът трябва да притежава следните характеристики:

1. Да бъде предсказуем - тази характеристика определя колко точно изходът от даден проект може да се предвиди преди проектът да е завършил. За целта се изследва самия проект и опитът от вече разработени проекти в организацията.
2. Да осигурява тестване и поддържане. Поддържането на даден проект струва значително, затова е необходимо да се разработва такъв софтуер, който е лесен за поддържане. Това, обаче, е трудна задача, защото разработчика и човекът, който ще поддържа софтуера не са от една организация. Поддържането не е фаза при разработването на даден продукт и следователно не се очаква разработчикът да се грижи за нея. Изследванията показват, че усилието се разпределя между фазите на разработване както следва:

Изисквания - 10%
Проектиране - 20%
Кодиране - 20%
Тестване - 50%

Какво показват тези данни? Противно на очакванията кодирането не е най-трудоемката дейност. Най-много труд изисква тестването, противно на общоприетото мнение, че то е странична дейност, която не е нужно дори да се планира!

Ето защо цел на процеса е да се намали цената на тестването и поддържането.

3. Да отстранява дефектите рано и да предпазва от тях

Грешки могат да се въведат по време на целия процес на разработване. Изследванията показват, че тяхното разпределение по фази е следното:

Анализ на изискванията - 20%
Проектиране - 30%
Кодиране - 50%

Относителната цена на отстраняването им не е една и съща. Колкото по-късно се открива една грешка, толкова по-скъпо е нейното отстраняване. Доказано е, че ако грешка от фазата анализ се коригира при тестването, цената за това е 100 пъти по-висока, отколкото ако се коригира в началото. Затова грешките в дадена фаза трябва да се откриват още преди тя да е завършила. Но още по-важно е процесът (методът) на разработване да предпазва от грешки, за да не се налага тяхното отстраняване. За целта се използва опитът от предишни проекти.

4. Да се самоподобрява

За целта процесът се анализира и се виждат неговите слаби места.

Процес за разработване

При този процес се обръща внимание върху дейностите, свързани с производството на софтуер - проектиране, кодиране, тестване. Той е основен за целия софтуерен процес. Състои се от отделни стъпки, във всяка от които се извършва добре дефинирана дейност, с оглед постигане целите на проекта. Изходът от дадена стъпка образува входа на следващата.

Процесът е просто и измеримо структурирано множество от дейности, извършвани с цел производството на специфичен резултат, който е предназначен за определен клиент или пазар. Докато продуктът показва какво се произвежда, при процеса ударението е върху това как се извършва определена дейност. Следователно процесът е „специфично подреждане на дейности във времето и пространството, с начало и край, ясно определени входове и изходи“. На тази база може да се определи и съдържанието на понятието процес за разработване на информационни системи.

Процесът за разработване е съвкупност от дейности, подредени във времето, чиято цел е създаване на информационни системи.

Процесът за разработване има общи и специфични характеристики. Общите му характеристики произтичат от това, че изграждането на информационни системи е вид производствен процес, а специфичните зависят от характера на създавания продукт.

Общите елементи, които са присъщи на всеки процес, са неговите вход, изход, последователност от дейности, цел, място и продължителност на изпълнение. При разработването на конкретни информационни системи (например банкови) тези елементи придобиват специфично съдържание: изискванията на банката са вход, а функциониращата система - изход от процеса на разработване.

Процесът за разработване на ИС се състои от дейности. Те се изпълняват в определена последователност. С цел по-лесното им управление дейностите се групират в *главни процеси*. Конкретният процес, който се изпълнява за реализиране на определена система, е екземпляр (проиграване) на процеса за разработване. В същия смисъл се използват и понятията поток от работа и нишка (thread) на процеса на разработване.

Понеже цел на процеса е да се открият дефектите във фазата където възникват, всяка стъпка завършва с верификация или валидация. Верификация означава тестване в края на всяка фаза. Пример: верифицира се проект, Спецификация. Валидация означава тестване след окончателното завършване на продукта и преди неговото приемане. Тя определя дали продуктът като цяло удовлетворява Спецификацията си. Следователно всяка фаза завършва с ясно дефиниран изход, който представлява вход за следващата фаза и може да се провери с някакви средства. Подобни междинни изходи в процеса на разработване се наричат работни продукти - например документа Спецификация, код на програма и др.

Макропроцес и микропроцес за разработване

Процесът на разработване се реализира посредством два главни процеса - макропроцес и микропроцес. Понятията макропроцес и микропроцес се утвърждават в теорията по информационни системи от Гради Буч.

Макропроцесът и микропроцесът се обособяват на база разделянето на дейностите на процеса на разработване на основни и поддържащи. Чрез поддържащите дейности се реализира управлението на процеса на разработване. Основните дейности са свързани с използването на конкретни методи и средства. Така при разработването на информационни системи се обособяват две нива - организационно и техническо. Организационното ниво има глобален характер и обхваща разработването на различни видове ИС. То дава най-общ поглед върху процеса на разработване и затова се нарича макропроцес. Техническото ниво е ориентирано към използваните компютърни методи и средства. То е подчинено на макропроцеса. Техническото ниво е силно зависимо от спецификата на произвеждания продукт, както и от индивидуалните

качества на разработчиците. Нарича се микропроцес.

Макропроцесът и микропроцесът за разработване се реализират паралелно. Макропроцесът засяга предимно стратегически аспекти на планирането и управлението на проекта. Той се реализира съвместно от потребителите и разработчиците. На негова основа се създава ред и организираност при разработването на ИС. От своя страна, микропроцесът обхваща някои тактически аспекти при изграждането на качествени системи. Той се реализира от разработчиците. В тази насока, макропроцесът може да бъде определен като управленски процес, а микропроцесът - като технически, насочен към изграждането на реални информационни системи.

Макропроцесът дефинира следните основни дейности за разработване на информационни системи (таблица): анализ, проектиране, реализиране, внедряване, използване и поддържане.

Основна дейност	Съдържание
Анализ на системата	Идентифициране на проблемите, които трябва да се решат чрез ИС. Определяне на възможните решения, избор на решение. Определяне на изискванията към новата система.
Проектиране на системата	Избор на информационни технологии. Определяне на методите и средствата за разработване. Модел на новата система
Реализация	Доставка/разработване на хардуера и софтуера; Конфигуриране на мрежата; Тестване; Интегриране
Внедряване	План за внедряване. Комплектоване на системна и потребителска документация.
Използване и поддържане	Работа със системата. Оценка. Отстраняване на грешки

Макропроцесът и микропроцесът са взаимосвързани. Макропроцесът насочва микропроцеса, като определя основните продукти, които трябва да се получат в резултат на изпълнението на определена дейност. Микропроцесът има различно проявление при специалистите, които извършват специфични дейности за реализиране на продуктите. От своя страна той е база за усъвършенстване на макропроцеса.

От особена важност за реализирането на успешни проекти е да се намери и поддържа правилен баланс между микро- и макропроцеса. Ако се подцени значението на микропроцеса, фокусът се измества върху организационните аспекти на процеса, а някои дейности, свързани с непосредственото производство на системата и индивидуалните качества на специалистите по информационни технологии (например прилагане на конкретен метод, програмиране на конкретен език, инсталиране на компютърна мрежа с определена конфигурация и т.н.), остават на втори план. При подценяване на макропроцеса е твърде вероятно да се получи резултат, различен от очаквания, т.е. система, която не удовлетворява целите и дефинираните изисквания.

Жизнен цикъл за разработване

Под жизнен цикъл се разбира съвкупността от етапи, през които минава разработването на програмния продукт (информационната система). Основните дейности на процеса на разработване обикновено се реализират последователно във времето. Често обаче се налага някои от тях да се повтарят неколкостранно. Възможно е също така някои дейности да се реализират паралелно във времето, а други да се пропуснат. Това се определя от

специфичните особености на разработваната система.

Последователността на дейностите определя логиката на процеса. На ниво макропроцес тя съответства на понятието модел на жизнения цикъл. Това понятие е утвърдено в практиката от американския учен Боем. Първоначално то се отнася само за разработването на софтуер. В последствие разширява своя обхват в областта на разработването на информационни системи и реорганизацията на управленски системи.

При жизнения цикъл за разработване основните дейности се обособяват в етапи. На тази база става възможно управлението и контрола върху тяхното изпълнение. Следва да се отбележи, че има различни мнения относно броя и наименованията на етапите на жизнения цикъл за разработване.

Процесът за управление на проекта

Подходящото управление е неделима част от разработването на софтуер. В голям софтуерен проект може да участвуват много хора за дълъг период от време. За да постигнат целите на разработването, а именно минимална цена, добро качество и спазване на график, ресурсите по проекта трябва да се разпределят по дейности.

Процесът за управление на проекта определя всички дейности, които трябва да се извършат за да се получи продукт с добро качество при минимални разходи. Основна задача на този процес е оптималното реализиране на избрания процес (метод) за разработване, което означава планиране. Процесът за управление на проекта се занимава с планирането, оценката на необходимите ресурси, създаването на график и неговото спазване, контролиране на проекта.

Дейностите на управление на проекта се разделят в три фази:

1. Планиране

Управлението на даден проект започва с планиране. Целта е да се разработи план за разработването на софтуера. Подходящото планиране е ключов фактор за успеха на проекта. Планът се разработва преди да започнат дейностите по разработването и се обновява по време на работата като се отчитат данните свързани с развитието на проекта. Основни дейности през тази фаза са: оценка на цената, определяне на графика и контролните точки, създаване на план за управление на качеството, създаване на планове за следене и контрол, набиране на персонал. Планирането на проекта е най-важната управленска дейност.

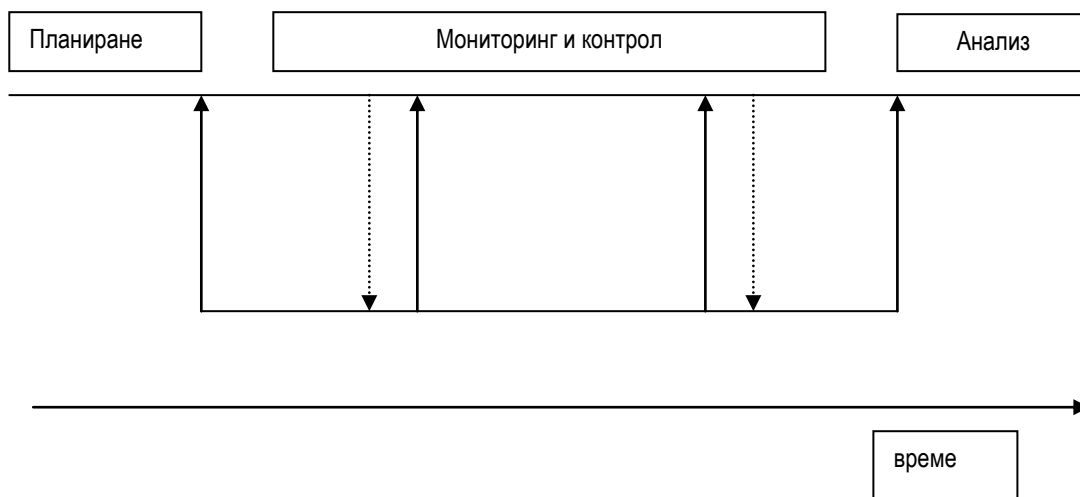
2. Следене и контрол

Това е най-продължителната фаза, защото обхваща по-голяма част от процеса на разработване. Следи се спазването на графика, бюджета на проекта и качеството. Освен това се следи за потенциални рискове, които биха попречили на проекта.

3. Анализ на приключването

Тази фаза настъпва след приключването на проекта. Целта е да се осигури информация за процеса на разработване. Подобна информация се събира от всички проекти като опит за да се осигури предсказуемост на процеса и оценка на бъдещи проекти.

Времевата връзка между процеса за управление и процеса за разработване е показан на фигура 2.2.



Фиг.2.2

На нея е показан идеалния случай, при който планирането се извършва преди за започне разработването, а анализът на приключването - след завършване на проекта.

За ефективното следене на проекта данните пристигащи от процеса за разработване трябва да са количествени и обективни. За да се постигне това се използват софтуерни метрики.

Софтуерните метрики са количествени "измерители, които измерват различни характеристики на софтуерната система. Всички инженерни дисциплини имат своите метрики - за тегло, за гъстота, дължина на вълната и др. Понеже софтуерът няма физически характеристики, стандартните метрики са неприложими. Затова са разработени специални метрики, които оценяват дължина, сложност и надеждност на даден софтуерен продукт.

Стойностите на някои метрики могат да се получат директно чрез измерване, докато стойностите на други се извеждат по косвен начин (наричат се индиректни метрики). Например някои от параметрите на качеството не могат да се измерят непосредствено.

За оценяването са нужни модели. Ако стойността на някои метрика не може да се получи директно, тогава се построява модел за нейната оценка като се използват стойностите на други метрики, които могат да се измерят. Моделът може да използва емпирични данни или може да бъде аналитичен.

Очевидно метриците, измерванията и моделите представляват едно цяло. Метриката дава количествена оценка на някое свойство, измерването дава действителната стойност на метриката за даден проект, а моделът осигурява стойностите на онези метрики, които не могат да се получат директно.

Всяка метрика трябва да има цел и обикновено тази цел се свързва с основната цел на проекта - ниска цена и високо качество. Следователно всяка метрика, която не допринася директно за подобряване на качеството или намаляване на цената е интересна единствено от академична гледна точка. Например за лингвистите или специалистите по кодиране е може би важно в една програма какви са честотите на отделните символи, но от гледна точка на софтуерното производство тя няма практическо приложение.

Процесът за управление на конфигурирането

По време на своето разработване софтуерът се състои от различни елементи - програми, данни, документация, които могат да се променят лесно. Затова е необходимо промените да се правят под контрол, т.е да се управлява конфигурирането на софтуера в процеса на разработване.

Дефиниция: Управление конфигурирането на софтуера е процеса на идентифициране и определяне на отделните елементи в системата, контролиране на промените в тях през целия им жизнен цикъл, отразяване на промените и проверяване на тяхната пълнота и коректност.

Процесът за управление конфигурирането на софтуера е независим от процеса на разработване, защото в модела на разработване не са предвиждат средства за управление на промените. В процеса на разработване се отчитат промени в рамките на дадена фаза - например програмиста непрекъснато променя програмата си докато я завърши, но не се отчитат промени в другите фази - например във фазата Изисквания. Подобни промени се контролират от процеса за управление конфигурирането на софтуера, който е във връзка с процеса на разработване. Да отбележим, че процесът за управление конфигурирането на софтуера директно контролира само продуктите на процеса за разработване и не оказва директно влияние върху самия процес.

Процесът за управление конфигурирането на софтуера се състои от три компонента:

1. Идентифициране на софтуерната конфигурация

Когато се прави дадена промяна е необходимо да се знае върху какво тя ще се прави. Продуктите от отделите фази, които са обект на промените се наричат базови продукти (baselines). Базовите продукти се определят формално след всяка фаза на процеса за разработване. Например след фазата Изисквания базов продукт е документа Спецификация, след фазата Проектиране базови продукти са отделните елементи на софтуера и техните проекти, разработената система представлява системен базов продукт и т.н.

Отделните базови продукти, които са обект на някакви промени не се разглеждат като неделими единици. Счита се, че те се състоят от известен брой конфигурационни елементи, като всеки конфигурационен елемент може да се разглежда като основна единица за промяна. По този начин базовият продукт може да се разглежда като подредено множество от отделни, точно определени конфигурационни елементи, които са със строго определена връзка помежду им. Всеки един от тези конфигурационни елементи може да има по няколко версии и проблемът е как и кои от тези версии да образуват базовия продукт, така че той да не е противоречив.

Основна задача на процеса за управление на конфигурирането на тази фаза е идентифицирането на конфигурационните елементи и установяване на зависимостите между отделните конфигурационни елементи. Конфигурационният елемент X зависи от конфигурационния елемент Y, ако промяна в Y налага промяна в X за да не се получи противоречив базов продукт.

2. Контрол на промените

При големи проекти всички решения относно промените се вземат от специална група наречена борд за контрол на конфигурацията, с ръководител Мениджър на конфигурирането. За малки проекти с контрола върху промени есе занимава един човек.

Обикновено когато даден конфигурационен елемент се разработва той се намира в работно състояние и не е достъпен за останалите конфигурационни елементи. Тогава той може да се променя свободно докато разработчикът остане удовлетворен. След това конфигурационният елемент се счита за замразен и отива за преглед, като разработчикът повече не може да го променя. Ако конфигурационния елемент е одобрен, той се записва в библиотека и вече е под контрол. Целта на този преглед е достигането на задоволително качество за всеки конфигурационен елемент. Ако даден конфигурационен елемент е незадоволителен, той се връща на разработчика за доработване.

След като конфигурационният елемент е въведен в библиотеката той не може да бъде повече променен без разрешение на мениджъра по конфигурирането. Ако той разреши промяна конфигурационния елемент се изважда от библиотеката, променя се, проверява се и се връща обратно в нея (управление на библиотеката). Обикновено старите версии не се изтриват за да не се пречи на другите разработчици, които работят с тях. В даден момент в библиотеката съществуват нови и стари версии и възниква задачата за управление на версиите.

3. Отчитане и ревизия (audit)

Понеже промените отнемат време са необходими механизми за отчитане на текущото състояние при конфигурирането на софтуера. Ревизията се провежда от независими лица. Целта им е да се установи дали се спазва процеса за управление конфигурирането на софтуера.