

Дәріс 1

Жобалау әдістер және программаның өміршендік циклын қамтамасыздандыру. Командалық және жеке жобалау және оған керекті аспаптар. Текстуальды және визуальды жобалау – технологиялары және әдістемесі, қолданатын аспаптары. Визуалды модельдеу дегеніміз не? **Визуальды модельдеу (visual modeling)** – нақты өмірдің объектілері мен ұғымдарын көрсететін және өзі көрініс табатын абстракциялардың көмегімен проблемаларды қабылдау әдісі. Модельдеу проблемаларын талдау, барлық қызығушылығы бар жақтар арасында ақпараттардың алмасуы (қолданушылар, пәндік аумақтағы мамандар, аналитиктер, дизайнерлер және т.б.), программалық қолданбалар мен мәліметтер базасын жобалау, және документацияны дайындау үшін пайдалы құрал болып табылады. Модельдеу талаптарын жақсы қабылдауға, жүйенің дизайн сапасын жақсартуға және оны басқару деңгейін жоғарылатуға өз септігін тигізеді. UML – объекті бағдарланған жүйелермен құрастырылатын мәндердің документациясын белгілеу мен көріністендіру үшін қолданылатын тіл.

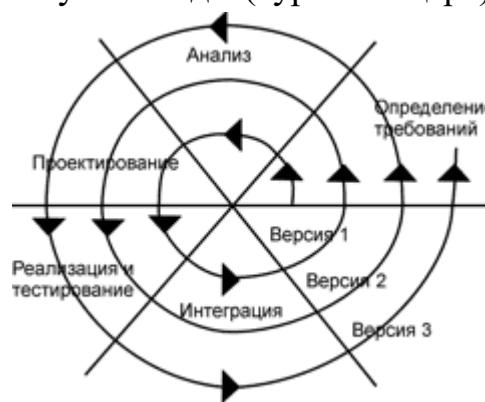
Модельдің көмегімен біз керекті жоқ детальдарды қарастырмай-ақ бар назарымызды негізгіге аудара отырып, проблеманы оңайлатуымыз мүмкін. Абстракциялау мүмкіндігі – күрделік феноменін шешуге көмектесетін, адам интеллектісінің іргелі қасиеті. Мыңдаған жылдар барысында суретшілер, қол өнершілер және құрылысшылар нақты шығармашылық ойларды жүзеге асыру алдында түрлі модельдерді жобалау талпыныстарын жасаған. Бұл құбылыстар программалық қамтаманы жасау индустриясын айналып өткен жоқ. Күрделі программалық жүйені құрастыру үшін, автор оның қасиеттерін түрлі көзқарастар тарапынан абстракциялауға, белгілеулердің нақты жүйелерінің көмегімен модельдерді жобалауға, олардың бастапқы талаптарға сәйкестігін тексеріп алуға міндетті, тек содан кейін ғана жүйені жаңа функциялармен толықтыра отырып, модельді практикада жүзеге асыруға болады.

12207 стандарты – Программалық жүйенің өміршендік циклының процесстері (ПЖ ӨЦ). Есептерді, жұмыстарды, процесстерді анықтайды. Кімге тағайындалғанын зерттейді. Нақты адаптация және шектеулер. Нормативті сілтемелер. Анықтамалар. Стандарттың қолданбалы қолдану – 5 негізгі, 8 қосымша, 4 ұйымдастыру процесстері. Өңдеу программасы – талаптардың талдануы, жобалау, программалау, жинау, тестілеу, орындау үшін енгізу, қабылдау. Этап бойынша жұмыстар тізімі. Этап бойынша жұмыстар мазмұны. Аспаптары.

Қазіргі заманғы аспаптық құралдар. Барлық өңдеу этаптарын визуализациялау. Визуальды технологиялар негізі. Өңдеу ортасы, объектілер, компоненттер кітапханасы. Аспаптар дайындау фирмаларының талдануы. Талданудың интерфейстік құрамы. Интерфейс стандарттары. Қазіргі заманғы визуальды аспаптардың құрамы. Программаларды өңдеу аспаптарының құрамын жобалаудың бірлік ұстанымдары және олардың әртүрлілігі.

RUP технологиясы, олардың өңдеу құжаттары және фазалары, аспаптары.

Rational Unified Process «Бүгінгі күнде ол ең көп кездесетін әдістемелердің бірі. Ол Rational Software компаниясымен өз өнімдерін қолдау үшін шығарылған, олар оннан көп деп есептеледі (ең көп кездесетін өнімдеріне - Rational Rose және Requisite Pro жатады). RUP үш атақты адамдармен құрылған - бұлар Гради Буч, Ивар Якобсон және Джеймс Рамбо (Rumbaugh). Олардың барлығы қиын программалық қамтаманы өндеуде үлкен жетістіктерге жеткен адамдар, оны RUP бейнесінен – ақ көруге болады. RUP **Итеративтілігі**, қазіргі заманғы дамыған кез – келген процесс сияқты, ол да итеративті болып табылады. Бұл яғни, құрылатын жоба бірнеше итерациямен орындалатындығын көрсетеді. Әрбір итерация аяғында жұмыс жасайтын өнім версиясын алуға мүмкіндіктер бар, бірақ олардың функционалдығы толық емес. Одан кейін орындалатын итерацияларда функционалдылық толықтырылып отырады да, соңында дайын өнім ала аламыз. Итеративті өнім идеясының процессін келесі түрде көрсетуге болады (сурет 2.1 қара).



Сурет.2.1 Спиральдің әрбір бөлігі - итерация. Яғни жүйе функционалдығы біртіндеп өсіп отырады.

Итеративті өндеудің артықшылықтары көп. Релиздердің көп бөлігі өнім сапасына ықпалын тигізеді, олар әрбір итерация сайын тестіленіп отырады. Яғни бастапқы тексерулер жүргізілгеннен кейін-ақ пайдаланушы күткен өнімді алуға мүмкіндіктер бар және өзгертулерді сол кезде енгізуге болады, егер ол қажет болса. Сонымен - қатар, жобаны жоспарлау да оңай, өйткені бірінші итерациядан кейін ақ ол түсінікті болады, және жобаны басқарушы келесі итерацияның аяқталу уақытын да алдын – ала жоспарлауға мүмкіндігі болады. Айта кететін жағдай, RUP-та процесстің итеративтілігін қатесіз жасауға болатындығы айтылмаған. Сондықтан RUP – ты бірінен кейін бірі орындалатын процесс үшін де қолдануға болады, яғни ондағы орындалу қадамдары бірінен-кейін-бірі орындалады, және дайын өнім соңында алынады. Сондықтан RUP орнатқанда оның итеративтілігіне және қатесіз орындалуына көңіл бөліп дұрыс ендіру керек. **Пайдаланушы сценаріі.** Пайдаланушы сценаріі (Use Case) – бұл белгілі бір операцияны орындау кезіндегі пайдаланушының әрекеттер тізбегінің сипаттамасы. Мысалы, жаңа құжат ашу үшін пайдаланушы сценаріін жазуға болады және т.б. RUP пайдаланушы сценарілерімен басқарылады (немесе прецеденттер). Пайдаланушы сценарілері өндеушілерге жүйе не жасауы және қалай жасауы керек екендігін нақты айтып отыру мүмкіндік береді.

Пайдаланушы сценарилері жүйенің функционалдық спецификациясының бөлігі болып табылады. Негізінен мұндай сценарийлер программаны өңдеу кезінде тигізетін пайдасы көп, өйткені олар:

- тапсырыс берушіге және жалпыға түсінікті тіл болып табылады да тапсырыс беруші мен өңдеуші екеуін байланыстырушы бөлім болып табылады;
- программа жұмысының логикасындағы қателерді бастапқы жұмыс барысында табуға көмектеседі;
- тапсырыс берушінің программаға деген талаптарын нақты табуға көмектеседі;
- текстілік сценарилерді жазу үшін интерфейс өңдеушінің базасы болып табылады.

RUP-тағы пайдаланушы сценарилеріне ерекше орын берілген, ол процесс use-case driven деп аталады, яғни пайдаланушының басқарушы сценарилері. **RUP құрылымы** Процесстің төрт фазасы бар: 1.Зерттеу (Inception) 2.Жоспарды нақтылау (Elaboration) 3.Құру (Construction) 4.Ашып қарау (Transition). Әрбір фазадағы негізгі көңіл әртүрлі процесстерге бөлінеді. Зерттеу фазасында талаптарды жинау және талдау жүргізіледі, Жоспарды нақтылау фазасында – жүйені жобалау және талаптарды талдау, құру фазасында - өңдеу және кодтау, ашып қарау фазасында – тестілеу және тарату. RUP әдістемесі **негізгі 9 ағынға** негізделіп жасалады: 1)Бизнес-талдау (керектіктің талдануы); 2) Талаптарды жинау және талаптарды басқару (талаптарды функционалдық спецификацияға ауыстыру); 3) Талдау және моделдеу (талаптарды программалық моделге ауыстыру); 4)Кодтау; 5)Тестілеу (программаның берілген талаптарға сәйкестігін тексеру); 6) Өзгертулерді және конфигурациясын басқару (өнімнің әртүрлі версияларындағы өзгерулерін тексеру); 7)Жобаны басқару ; 8) Өңдеу ортасын ұстану және құру; 9) Ашып қарау (өнімді беру немесе сату үшін керектінің барлығы). RUP – тағы кез –келген өнім төрт фазадан өтеді. Осы фазалар арқылы барлық тоғыз ағын да өтеді. Әрбір фаза, өз кезегінде, итерацияларға бөлінеді. Егер, мысалға алатын болсақ, "Зерттеу" фазасындағы бірінші итерация болса, онда бұл итерациядағы негізгі көңіл бизнес-талдауға, талаптарды жинау мен моделдеуге, және кодтауға бөлінеді. Егер "Құру" фазасындағы соңғы бір итерациялардың бірін алтын болсақ, онда негізгі көңіл кодтауға, тестілеуге және конфигурацияны басқаруға бөлінеді. Басқаша айтсақ, жобаның дамуына байланысты әрбір итерациядағы кемшіліктер жойылады. Бұл әрине дұрыс, соңына қарай талдау мүлде керек болмайды, ал талаптарды жинауға кеш болады. **Артефакт (Artefact)** деп өнімді атаймыз, ол ПҚ өңдеу процесі кезінде құрылады және қолданылады. Мысалы, артефакт болып құжаттар, моделдер, орындалатын кодтар алынады. Артефакт мысалдары: пайдаланушы құралы, UML-дағы класстар диаграммасы және т.б. RUP-тың бөлінбейтін бөлігін артефакттар мен олардың атқаратын қызметтері толықтырады. Программаны өңдеу кезінде әртүрлі артефакттар құрылады, сол немесе басқа артефакттардың құрылуына оның атқаратын қызметтері жауапты болады. Мысалы, класстар диаграммасын "Архитектор" құрады, ал сценарилеін және тестіленуін "Тест дизайнерлері" жазады. Барлық визуальды моделдеу CASE-құралдарының көмегімен жүзеге асады. Оның негізін UML тілі құрады

(Unified Modeling Language), ол таңқаларлық жағдай емес, өйткені UML RUP авторларымен ойлап табылған дүние. **Ең жақсы практикалар** RUP-тың өзі ең жақсы алты практикаға негізделген (best practices):

- Итеративтік өңдеу
- Талаптарды басқару
- Модульдік архитектураны қолдану
- Визуальды моделдеу
- Сапасын тексеру
- Өзгертулерді бақылау

Олар RUP-тың бөлінбейтін бөлігі болып табылмайды, бірақ оларды процессті құру кезінде ұстанған дұрыс деп есептелінеді.

Итеративтік өңдеу бастапқы стадияларда жұмыс жасайтын дайын өнім версиясын алуға мүмкіндік береді және қателерін табуға болады, сонымен қатар, нәтижесінде арынған өнімнің сапасы да жақсы болады, өйткені база өнімде қанша итерация болса сонша рет тестіленеді.

Талаптарды басқару – ол қиын немесе қиын емес өнімдерді алу кезіндегі маңызды бір процесстердің бірі. Бұған байланысты өнім, тапсырыс берушінің талаптарына сәйкестендіріледі. Аспаптық қастамасыздандыру Requisite Pro – ның көмегімен шешіледі.

Теория жүзінде модульдік құрылым кодты қайта қолдануға болады, және жүйе иілгіш болады деп айтылған. Ал практика жүзінде мұны тарату мүмкін емес.

Визуальдык моделдеу жүйенің өсетін қиындықтарымен тиімді күресуге мүмкіндік береді. Моделдер негізінен, жүйенің не істейтіндігін және қалай істейтіндігін түсіну үшін пайдаланылады. Сонымен қатар, моделдер өңдеушілердің арасындағы коммуникация құралы болып табылады, бірік ол үшін ол барлығына да түсінікті болу керек. Міне осы үшін RUP-та UML қолданылады, ол өңдеушілерге бір тілде сөйлеуге мүмкіндік береді. Аспаптық қолдау Rational Rose – пен қамтамасыздандырылады.

Өнім сапасы – бұл да оның маңызды мініздемесінің бірі. Айтылғандай, RUP – мүмкіндігінше сапа дейгейіне бағытталға, бірақ адаптация процессінде, егер адаптация сәтті болмаса, онда бұл сапамен байланысты әдістемеді проблемалар тууы мүмкін. Аспаптық ұстанымдар бірнеше программалармен қамтамасыздандырылуы мүмкін олар: Rational Purify, Rational PureCoverage, Rational Quantify, Rational Robot.

Өзгерулерді бақылап отыру тапсырыс берушінің талаптарының өзгерулеріне және сыртқы құрылғы өзгерістеріне тез жауап беруге мүмкіндік береді. RUP-тың процесстері болады, олар өзгерулерді тиімді бақылауға мүмкіндік береді. Аспаптық ұстанымдар келесі программалармен қамтамасыздандырылуы мүмкін олар: Rational ClearCase және Rational ClearQuest.

RUP қалыптастыру. RUP адаптацияланатын процесс болып табылады, яғни оларды керекті бір команданың немесе керекті бір жобаның пайдалануына байланысты құрауға болады. Сонымен қатар, мұны жасау керек те, өйткені

мұнсыз RUP-ты пайдалану тиімділігі нөлге талпынады. RUP 2000-да, мысалы, 30-дан астам ролдер және 50-ден астам артефактар бар. Шындығында, егер команда 5 адамнан тұрса, онда ондағы барлық ролдерді шығару және барлық артефакттарды құру мүмкін емес. Негізі команда неғұрлым аз болса, соғұрлым процесс жеңіл болады. Яғни минимум артефакттар құрып, минимум ролдерді енгізу керек.

Сонымен RUP процессті қалыптастыруға мүмкіндік береді. RUP-тың жалпы сипаттамасынан сапалы өнім шығару үшін және бюджет аймағында болатын, командада қолданылатын керек процесстер, ролдер және артефакттарды алу керек. Мысалы, талаптарды басқаруды алатын болсақ. RUP-тың жалпы сипаттамасында келесі артефакттар бар: талаптарды басқару жоспары, пайдаланушы сценарилерінің моделі, жүйенің талаптарының спецификациясы, көріністер, талаптар репозиторийі. Кішкентай жоба үшін талаптар спецификациясы да және жеке сценарилер де жеткілікті. Ал үлкен жоба үшін репозиторий міндетті түрде қажет, өйткені онсыз талаптарды өзгертулер мен оларды бақылау жүйелік аналитиктің жұмысын қиындатады. Бірақ RUP-ты бір нақты жоба үшін қалыптастыру – нетривиальды есеп. Егер ол жұмысты бұрын RUP енгізумен жұмыс жасамаған адам жасаса, онда орындалатын есепті алу мүмкін емес. Негізі, RUP өте қиын әдістеме деп есептелінеді, оны негізінен үлкен командалар үшін қолданған жақсы. RUP-тың жұмысын жеңілдететін бір әдістемелердің бірі бұл dX. Rational Unified Process (RUP) әдістемесі келесі процесстерден тұрады (Workflows):

- бизнес-моделдеу;
- талаптарды басқару;
- талдау және жобалау;
- тарату;
- тестілеу;
- ашып қарау;
- конфигурация және өзгерулерді басқару;
- жобаны басқару;
- аспаптық қолдау.

RUP — итерациондық әдістемесі. Итерациондық әдіс есепті дұрыс түсіну үшін пайдалы, ол есептегі әрбір итерациялар арқылы біртіндеп қателерді табу және оларды жоюды орындауға мүмкіндік береді. Итерацияларды талаптарды дұрыс орындаған кезде және өзгертулерді бақылаған кезде ғана ұйымдастыруға болады. RUP моделді өңдеу және қолдауға енгізілген, ал қағаз құжаттар құру үшін ол керек жоқ.

Таңдалып алынған құрылым программалық компоненттердің өңделуін жоспарлау үшін қолданылады. Өнімді өңдеу негізінен жүйемен қолданылатын сценарилерді анықтаудан басталады (Use cases). Сценариилер процесстің барлық өміршеңдік циклын бағыттайды (бизнес-моделдеу, талаптарды таңдау, талдау және жобалау, тестілеу) және жүйені ашып қарау кезінде есептің орындалуын қамтамасыздандырады.

RUP объектілі-бағытталған технологияны қолдайды. Көптеген визуальды моделдер и объектілі-бағытталған модель болып табылады, олар объект

концепциясына, класстарға және олардың арасындағы байланыстарға негізделеді. Бұл кездеге жалпы тіл ретінде Unified Modeling Language (UML) алынады.

RUP жүйенің компоненттік өңделуін қамтамасыздандырады. Компонент деп тривиальды емес модулдер саналады. RUP жобада құрылатын барлық материалдардың сапансын бақылауға бағытталған, құрылатын жүйенің сапасына жауапты болып табылады. Сапаны бақылау бағасы процесстің әдістемесінде берілген.

RUP негізінен қиын ақпараттық жүйелерді (АЖ) құру үшін қолданылады, өндіріс масштабын және аспаптық құралдарды Rational Software қолдайды, ол жобамен жұмыс кезіндеге командалық жұмысты қамтамасыздандырады.

Программалық өнімді өңдеу кезінде қолданылатын мемлекеттік және шетелдік стандарттар. Өңдеу сапасын анықтайтын ӨАҚ стандарты. ИСО 4001-96. Сапа жүйесі. Жобалаудағы сапаны қамтамасыздандыру моделі. ӨАҚ МЭК 9126-93. АТ. Программалық Өнімнің бағасы. Сапа мінездемесі және оны басқарудағы қолдану құралдары. ӨАҚ МЭК 8402-94. Сапаны басқару және сапамен қамтамасыздандыру. Сөздік. 34.601-90, 34.603-92, ӨАҚ 4001-96 стандарттары және т.б.

Аспаптық құралдарға стандарттардың таратылуы. Өңдеудің визуальды элементтерінің жиынын дайындау. Элементтер кітапханасының құрамын жедел құрастыру мүмкіндіктері. Жобалаудың жетіспейтін стандартты элементтерін қосу. Өзгертудің оперативтігі.

Ақпаратты қорғау стандарттары. P511.88-98. Ақпаратты қорғау. Программалық құралдарды компьютерлік вирустардың бар жоқтығына тексеру. P512.41- 98. Қатынауды басқару, бақылау жүйесі және құралдары. Классификациясы. Жалпы техникалық талаптар. Тексеру әдістері. Кодтың комбинация мәндері – санкцияланбаған қатынаудан қорғау тұрақтылығы. Қатынауды басқарудың жабдық программасы не істеуі керек? Программаның әрекеттерге тұрақтылығы.

Негізгі әдебиеттер: 1 нег. [7-57], 5 нег. [7-15];

Қосымша әдебиетер: [1] – 176, [5] -25 б

Бақылау сұрақтары:

1. Программаны өңдеу процесінің құрамын және тәртібін негізгі қандай ГОСТ-лар анықтайды?
2. Пәнді оқып үйренуді қалай ұйымдастыруға болады?
3. Программаны өңдеу процесінің этаптары?
4. Жоғарғы деңгей қорғанысындағы кілттер жиынына қатынау талаптарын атаңыз?
5. Өңдеу сапасын қамтамасыздандыру мінездемесі қандай?