

## 10-семинар сабағы. Бағдарламалық жабдықтаудың сапасына қол жеткізу

Сапалы Бағдарламалық жабдықтауларға қалай қол жеткізуге болады?

Қадамдық нұсқаулық

### САПАНЫ ҚАМТАМАСЫЗ ЕТУ ЖОСПАРЫ: ОҒАН НЕ КІРЕДІ?

Жалпы сапа мазмұны дәріс сабағында қаралды.

Әдетте, бағдарламаны өңдеушілер (программисттер) өздері жасаған бағдарламалық жабдықтаудың керемет сапаға ие болатынына күмән келтірмейді. Алайда, бұл шынымен де болуы үшін, жоба басталғанға дейін сапаны қамтамасыз ету жоспарын құру және бағдарламалық жасақтаманың жақсы сапасы туралы келісу маңызды. Бұл материал жоба менеджерлеріне және АТ жобаларын жүзеге асыруға жауапты барлық адамдарға ұсынылады.

АТ жобасын жоспарлау кезінде не ескеріледі? Тапсырыс берушінің талаптары, жұмыс көлемі, бюджет, мерзімдер. Сондай-ақ, ең тиімді құралдар мен қажетті дағдылары бар мамандарды таңдау маңызды.

Ештеңені ұмытпаған дұрыс, оның ішінде бастысы - сапаны. Оны да жоспарлау керек. Шыныменде, сапа уақыт, бюджет, адами ресурстар мен технологиядан кем емес. Егер оны жоспарламасақ, жобаның кейінгі кезеңдерінде қиындықтар туындауы мүмкін, оларды жою оңай болмайды.

Әр жобаның іс-қимыл жоспары бар. Сонымен қатар, әр жобада міндетті түрде сапаны қамтамасыз ету жоспары (SQA жоспары) болуы керек. Өкінішке орай, бұл жоспардың қалай көрінетінін және оған не кіру керектігін білетіндер аз.

Көбінесе SQA жоспары тестілеу жоспары болып саналады. Біз сізді сендіруге асығамыз: бұл екі түрлі нәрсе, бірақ олар бір-бірімен тығыз байланысты. Тестілеу жоспары тестілеу іс-шараларының стратегиясын сипаттайды, ал сапа жоспары клиенттердің күтулерін, қабылдау критерийлерін, сапаны жоспарлы бақылау және **технологиялық аудит** жұмыстарын, өзгерістерді басқару процедураларын қамтуы керек.

Жоба қабылданған мақсаттарға сәйкес келуі үшін сізде Тапсырыс беруші де, тестілеу тобы да мақұлдаған сапа жоспары бар екеніне көз жеткізіңіз.

**МҮНДАЙ ЖОСПАРДЫ ЖАСАУ ҮШІН ҮШ ҚАДАМДЫ ОРЫНДАУ КЕРЕК.**

### **1. Жақсы сапа өлшемдерін анықтау**

Не жақсы, не жаман екенін анықтаудың ең оңай жолы - алдыңғы тәжірибені талдау. Мысалы, ол клиенттің наразылығын тудырды, пайдаланушылар форумдар мен қорларда шағымданды. Мүмкін ол бағдарламалық өнім өте баяу болған шығар? Немесе онда тым көп ақаулар бар ма? Немесе ондағы төлемдер қауіпті болды ма? Осы сұрақтарға жауап беріп және барлық маңызды аспектілердің тізімін жасау керек (өнімділік, пайдалану ыңғайлылығы, қауіпсіздік).

### **2. Өлшенетін сапа мақсаттарын қою**

Шынында да, бюджет ақшалай түрде, мерзімдер нақты күндерде өлшенеді. Сапаны қалай өлшеуге болады? Қазір айтамыз.

Өнім сапасының барлық маңызды аспектілерінің тізімін жасағаннан кейін, әрқайсысын қалай өлшеу керектігі туралы ойлану керек.

Мысалы:

Егер ақаулардың саны маңызды болса, "ақаулардың тығыздығы" (defect density) терминін қолданамыз. Ақаулардың тығыздығы - компоненттің немесе жүйенің өлшеміне бөлінген компоненттегі немесе жүйедегі ақаулар саны (нақты өлшем бірліктерінде, мысалы, код жолдарында көрсетілген).

Егер өнімділік маңызды болса, бағдарламалық жабдықтаудың жауап беру уақыты (секундпен көрсетілген), ақпаратты беру жылдамдығы (секундына байт) немесе жүктеме (жүйені бір уақытта тиімді пайдалана алатын пайдаланушылар саны) туралы сөйлескен жөн.

Сіз үшін маңызды Сапа өлшемдерін анықтай отырып, нақты, қол жетімді мақсаттар қойыңыз, олар жобамен жұмыс жасау кезінде сіздің бағдарыңыз болады.

Сізде мақсат қою тәжірибесі бар ма? Онда сіз оған сенімді бола аласыз. Алайда, көбінесе келесі жобаны жоспарлай отырып, менеджерлер бұрын қол жеткізілген көрсеткіштерді жақсартқысы келеді. Егер бұл сіздің жағдайыңыз

болса, онда өткен жобаның өлшемдерін өлшеп, оң өзгерістермен жаңа мақсаттар қою керек..

### **3. Сапа бойынша жұмысты жоспарлау**

Сапаға қойылатын талаптарды анықтау жұмыстың соңы емес. Берілген мақсаттарға жету үшін әрекет қажет.

Өнімнің белгілі бір сапасына қол жеткізудің үш түрі бар: **ақауларды анықтау, ақауларды түзету және олардың пайда болуына жол бермеу.**

Ақауларды анықтау әрекеттері жазбаша кодтағы қателерді табуға және оқшаулауға бағытталған. Тестілеумен айналысатын команда ақаулар мен оларды жөндеуді тексереді. Бірақ тестілеу тым кеш болмаса.

Келесі механизмдер арқылы тестілеу шығындарын азайтуға тырысу керек:

Дизайнды талдау (design review) және жазбаша кодты талдау (code review). Кодты талдау қосымшаның маңызды орындары үшін өте маңызды (аутентификация механизмдері, қаржылық операцияларды өңдеу және т.б.). Бұл тәжірибенің айқын артықшылықтары: кодтың жалпы сапасы жақсарады, код бір стильге келеді, ақымақ қателер мен терулер анықталады.

**Test-Driven Development (тестілеу арқылы әзірлеу).** Осы тәсілмен белгілі бір функцияны жазу алдында қабылдау тесттерінің жиынтығын жасау керек. Дамудың бұл тәсілі жоғары сапалы кодты шығаруға, сондай-ақ осындай кодты алып жүру кезінде туындайтын көптеген мәселелерді жоюға мүмкіндік береді.

Ақауды түзету үшін кодты жазған әзірлеуші жауап береді. Ақаулықты түзету туралы есептерден кейін тестілеу мамандары өнімнің басқа аймақтарына түзетулер енгізілмегеніне көз жеткізу үшін регрессиялық тестілеу жүргізеді.

Соңғы (тәртіппен, бірақ маңызды емес) белсенділік ақаулардың пайда болуына жол бермейді. Бенджамин Франклин айтқандай, ауруды емдегеннен гөрі, алдын алған оңай деп.

Проактивті жоба сапасын басқару теорияда ғана емес, тәжірибеде де өзінің тиімділігін әлдеқашан дәлелдеген. Мүмкін болатын ақауларды болжауға не көмектеседі? Бағдарламалық жасақтама талаптарын мұқият талдау, ықтимал тәуекелдерді талдау, дамудың ең тиімді әдіснамасын таңдау,

құралдар жиынтығы, Тапсырыс берушінің талаптары мен пайдаланушының күтулерін құжаттау.

Әрине, қосымшаны жобалау сатысында тәуекелдерді сауатты талдау үшін мамандарға жүгінген дұрыс. Бизнес-сарапшылар мен сапа бойынша кеңесшілер кодты жазбас бұрын сенімді қолдау көрсетеді және жөндеу шығындарын болдырмауға көмектеседі.

БКС