

5 PROJEKTMENEDZSMENT

TÉMAKÖRÖK

Ennek a fejezetnek az a feladata, hogy betekintést adjon a projektvezetésbe. Elolvasva a fejezetet, az alábbiakkal leszünk tisztában:

- megismerjük a szoftverprojekt menedzselésének alapvető feladatait;
- megértjük, miért nehezebb menedzselni a szoftverprojekteket, mint más mérnöki projekteket;
- megértjük, miért elengedhetetlen a projekttervezés minden egyes szoftverprojektben;
- megértjük, hogy hogyan tud grafikus reprezentációkat (oszlopdíagram, tevékenységdiagram) használni a projektmenedzser a projekt ütemezésének reprezentációjára;
- bevezetőt kapunk a kockázatkezelés folyamatába, és megismerünk néhány kockázati tényezőt, melyek előbukkanhatnak egy szoftverprojektben.

TARTALOM

- 5.1. Vezetői tevékenységek
- 5.2. A projekt tervezése
- 5.3. A projekt ütemezése
- 5.4. Kockázatkezelés

A szoftverprojekt-menedzsment fontos része a szoftvertervezésnek. A jó menedzsment persze nem garantálja még a projekt sikerét. Mindazonáltal a rossz menedzsment általában a projekt sikertelenségét okozza. A szoftvert későn szállítják le, a költségek meghaladják az eredetileg tervezett költségeket, és az is előfordulhat, hogy nem fognak megfelelni a követelményeknek.

A szoftvermenedzsment felelőssége megtervezni és ütemezni a fejlesztési projektet. Ők felügyelik a munkát, hogy biztosítsák, az a szükséges szabványok szerint van végrehajtva. Megfigyelik és folyamatosan ellenőrzik, hogy a fejlesztés megfelelő időterv és költségvetés szerint halad. A jó menedzselés nem tudja garantálni a projekt sikerességét, azonban a rossz menedzselés a legtöbb esetben

a projekt bukását okozza. A szoftvert későn szállítják le, a költségek az eredeti becslések többszöröseire rúgnak, és a rendszer a követelményeknek sem felel majd meg minden tekintetben.

A szoftvermenedzserek ugyan azt a fajta munkát végzik, mint egyéb más mérnök projektmenedzserek, azonban a szoftvertervezés számos tekintetben különbözik minden más egyéb típusú tervezéstől, így a szoftver menedzselése eléggé nehezzé válik. Íme néhány ilyen különbség:

1. *A szoftver nem kézzelfogható.* Egy hajóépítészeti projekt menedzsere vagy egy építőmérnöki projekt menedzsere látja a terméket annak fejlesztése alatt. Ha a munka ütemezése elcsúszik, az látszik a terméken is. A szerkezet részei nyilvánvalóan nem lesznek befejezve. De a szoftver megfoghatatlan. Nem látható vagy tapintható. A szoftverprojekt menedzserei nem látják annak haladását. Csak arra támaszkodhatnak, hogy a projekt átvizsgálásához szükséges dokumentációt mások előállítják.
2. *Nincsenek szabványos szoftverfolyamatok.* Még nem értettük meg teljes mértékben a szoftverfolyamatok és a terméktípusok közötti kapcsolatot. A mérnöki tudományágakban a folyamat már történelmileg is hosszú ideje kipróbált, tesztelt. A különleges típusú rendszerekhez, mint például a hidakhoz tartozó tervezői folyamatok jól ismertek. A mi felfogásunk a szoftverfolyamatról az elmúlt pár évben jelentősen átalakult, fejlődött, mégis, még mindig nem tudjuk nagy bizonyossággal előre megmondani, a szoftverfolyamat nagy valószínűséggel mikor botlik fejlesztési problémákba.
3. *A nagy szoftverprojektek gyakran több „különálló” projekt.* A nagy szoftverprojektek gyakran különböznek minden más korábbi projekttől. A menedzsereknek ezért sok korábbi tapasztalatot kell szerezniük, amelyek alapján csökkenteni tudják a projekten belüli bizonytalanságokat. Következésképpen sokkal nehezebb előre felkészülni a problémákra. Ráadásul a gyors technológiai váltások a számítástechnika és kommunikáció területén a korábbi tapasztalatokat elavulttá teszik. A leckét megtanultuk, a tapasztalatok nem mindig vihetők át az új projektekre.

Ezekből a problémákból adódóan nem meglepetés, hogy számos szoftverprojekt késik, túllépi a költségvetést és a határidőket. A szoftverrendszerek gyakran újak és technikailag innovatívak. Az innovatív tervezői projektek (például új szállítási rendszer) szintén gyakran küszködnek határidőkből fakadó problémákkal. Látva az érintett nehézségeket, talán figyelemre méltó, hogy léteznek projektek, melyek a megadott időn és költségvetésen belül elkészülnek.

A szoftverprojekt-menedzsment szerteágazó téma, túllépi egy könyvfejezet kereteit. Ez a fejezet bevezet a témakörbe, és áttekintünk három fontos menedzsmenttevékenységet, nevezetesen a projekttervezést, a projektütemezést és a kockázatkezelést. A későbbi fejezetek (6. rész) más szoftvermenedzselési aspektusokkal foglalkoznak, figyelembe véve a szoftver költségbecslését és a minőség menedzselését.

5.1. VEZETŐI TEVÉKENYSÉGEK

Nagyon fontos, hogy a szoftver menedzseléséhez szabványos munkameghatározást írjunk. A munkák erősen függnek a szervezetektől és a fejlesztés alatt álló szoftverterméktől. Azonban minden egyes menedzsernek kötelezettsége bizonyos szakaszokban az alábbi tevékenységekből néhányat vagy akár valamennyit elvégezni:

- indítványokat írni,
- a projektet megtervezni és ütemezni,
- a projekt költségét figyelemmel kísérni,
- a projektet figyelni és felülvizsgálni,
- a személyzetet kiválasztani és fejleszteni,
- beszámolókat írni és előadni.

A szoftverprojekt első szakaszában szokott sor kerülni a projekt végrehajtásával foglalkozó indítványok megírására. Az indítványok a projekt céljait írják le, és hogy azok hogyan érhetők el. Gyakran tartalmazznak költség- és ütemezési becsléseket is. Megindokolhatják, miért ítéltető oda egy projekt szerződése egy konkrét szervezetnek vagy csapatnak. Az indítványok írása kritikus feladat, ugyanis számos szoftverszervezet létezése függ a megfelelő mennyiségű elfogadott indítvány és odaítélt szerződés meglététől. Nem igazán lehet útmutatót adni ehhez a feladathoz. Az indítványok írásához olyan képesség kell, ami csak tapasztalatok árán megszerezhető.

A projekttervezés a projekt tevékenységeinek azonosításával, a mérföldkövek és a leszállítható részeredmények biztosításával foglalkozik. Fel kell vázolni a tervet, amely a fejlesztést végigvezeti a projekt céljai felé. A költségbecslés kapcsolódó tevékenység, melynek tárgya a projektterv valóra váltásához szükséges erőforrások becslése. Ezekkel részletesebben foglalkozom a fejezet későbbi részeiben, illetve a 26. fejezetben.

A projekt figyelése folyamatos projekttevékenység. A menedzsernek nyomon kell követnie a projekt előrehaladását, és összehasonlítani az aktuális és a tervezett haladást, valamint a költségeket. Habár a legtöbb szervezetnek létezik erre valamilyen formális mechanizmusa, gyakran csak a szakavatott menedzser képes tiszta képet formálni arról, mi folyik ténylegesen a projektben, a projekt tagjaival történő informális beszélgetések segítségével.

A nem formális figyelés gyakran előre megmondja a projekt potenciális problémáit, és hogy azok milyen nehézségeket okozhatnak. A tagokkal történő napi megbeszélés feltárhat konkrét problémákat, például egy szoftverhiba keresésével kapcsolatban. Ilyenkor ahelyett, hogy egy ütemezési pontra kellene várni, ahol azt bejelentik, a szoftvermenedzser odarendelhet néhány szakértőt a problémához, vagy eldöntheti, hogy a problémát kerüljék-e ki.

Egy projekt során teljesen normális dolog, hogy van egy sor formális projektmenedzselési ellenőrzés, felülvizsgálat. Ezek a projekt átfogó előrehaladási és technikai fejlődését érintik, és összevetik a projekt aktuális állapotát a szoftvert kivitelező szervezet céljaival is.

Egy felülvizsgálat kimenete akár a projekt idő előtti befejezését is eredményezheti. Egy nagy szoftverprojekt fejlesztési ideje akár több év is lehet. Ez alatt az idő alatt a szervezeti célok majdnem mindig megváltoznak. Ezek a változások azt is jelenthetik, hogy a szoftverre nincs tovább szükség, vagy a projekt eredeti követelményei már nem állják meg a helyüket. A menedzsmentnek ekkor dönteni kell arról, hogy abbahagyják-e a szoftver fejlesztését, vagy megváltoztatva a projektet, azt hozzáigazítják a szervezet megváltozott céljaihoz.

A projektmenedzsereknek a legtöbb esetben ki kell válogatniuk azokat az embereket, akik a projekten fognak dolgozni. Ideális esetben a megfelelő tapasztalattal rendelkező szakavatott emberek rendelkezésre állnak a projekthez. Viszont a legtöbb esetben a menedzsereknek be kell érniük az ideálisnál rosszabb projektcsapattal is. Ennek okai a következők:

1. A projekt költségvetése nem teszi lehetővé, hogy jól fizetett munkaerőt alkalmazzunk. A kevesebb tapasztalattal rendelkező, kevésbé jól fizetett munkaerőt kell alkalmaznunk.
2. Nem áll rendelkezésre a megfelelő tapasztalattal rendelkező munkaerő sem a szervezeten belül, sem azon kívül. Ilyenkor lehetetlenné válik új munkaerő toborzása a projekthez. A szervezeten belül lehet, hogy a legjobb emberek már más projekten dolgoznak.
3. A szervezet úgy kívánhatja, hogy beosztottainak szaktudását fejleszteni kell. Tapasztalatlan munkaerőt rendelhetnek a projekthez, hogy tanuljon, és növelje tapasztalatait.

A szoftvermenedzszernek ilyen megszorítások mellett kell dolgoznia, mikor kiválogatja a projekt munkatársait. Viszont ha egy projekttagnak sincsenek tapasztalatai az adott típusú rendszer fejlesztésében, akkor nagy a valószínűsége a problémák bekövetkezésének. A megfelelő tapasztalat hiányában számos kis hiba vétető a fejlesztés közben. A csapatok felépítéséről és a munkatársak kiválasztásáról a 25. fejezetben lesz szó.

A projektmenedzszernek gyakran kötelezettsége az is, hogy tudósítson a projektről mind a kliens, mind a vállalkozó szervezet felé. A projektmenedzsereknek tömör, egyértelmű dokumentumot kell készíteniük, amely a részletes projektbeszámoló kritikus információit tartalmazza kivonatokban. Ezeket az információkat prezentálniuk kell a folyamatos felülvizsgálatokon. Következésképpen hatékony kommunikációs készséggel kell rendelkezniük mind a beszéd, mind az írás területén, ugyanis ez elengedhetetlen szaktudás a projektmenedzserek számára.

5.2. A PROJEKT TERVEZÉSE

A szoftverprojekt hatékony menedzselése nagyban függ a projekt előrehaladásának alapos megtervezésétől. A projekt vezetésének előre látnia kell az esetlegesen felmerülő problémákat, és hozzávetőlegesen megoldásokat kell előkészítenie az

adott problémákra. A projekt indulásakor felvázolt tervet úgy érdemes használni, mint a projekt egyik irányvonalát. Ennek a kezdeti tervnek a lehető legjobb tervnek kell lennie, amely a rendelkezésre álló információkból adható. Természetesen ez fokozatosan továbbfejlődik, ahogy a projekt előrehalad és több, jobb információ válik elérhetővé.

Az 5.2.1. alfejezet a szoftverfejlesztési terv szerkezetét tárgyalja. Mint a projektterv esetében, a menedzsernek egyéb típusú terveket is fel kell vázolnia. Ezek tömör leírása látható az 5.1. ábrán, amelyek azután másutt, a könyv hozzájuk tartozó fejezetében részletesebben kifejtésre kerülnek.

Terv	Leírás
Minőségi terv	Meghatározza azokat a minőségi eljárásokat és szabványokat, amelyeket a projektben használni kell (24. fejezet).
Validációs terv	Meghatározza a megközelítési módot, az erőforrásokat és az ütemterveket, amelyeket a rendszer validációjához használni kell (19. fejezet).
Konfigurációkezelési terv	Meghatározza a konfigurációkezelési eljárásokat és szerkezeteket, amelyeket alkalmazni kell (29. fejezet).
Karbantartási terv	Előre megadja a rendszer karbantartási követelményeit, a karbantartási költségeket és a szükséges erőfeszítéseket (27. fejezet).
Munkaerő-fejlesztési terv	Meghatározza, hogyan kell a projektcsapat tagjainak szaktudását és tapasztalatait fejleszteni (22. fejezet).

5.1. ábra. A terv típusai

Az 5.2. ábra által mutatott pszeudokód a szoftvertervezéshez használt projekttervezési folyamatot írja le. Azt mutatja meg, hogy a tervezés iteratív folyamat, mely csak akkor válik teljessé, ha maga a projekt is teljessé vált. Ahogy a projekt információi elérhetővé válnak a projekt ideje alatt, a tervet rendszeresen át kell vizsgálni. Az üzlet átfogó céljai nagyon fontos tényezők, melyeket figyelembe kell venni, mikor a projekttervet megfogalmazzuk. Ahogy ezek megváltoznak, úgy szükségképpen a projekttervet is meg kell változtatni.

A tervezési folyamat a projektet meghatározó megszorítások összegzésével indul (megkívánt szállítási határidő, rendelkezésre álló munkaerő, teljes költségvetés stb.). Ezekkel a projekt olyan paramétereinek becslésével együtt kell foglalkozni, mint például annak szerkezete, mérete, funkcióinak eloszlása. Az előrehaladás mérföldköveit és a részeredményeket csak ezután határozzuk meg. Ezek után a folyamat belép egy ciklusba. A projekt ütemtervét fel kell vázolni és az ütemterv által meghatározott tevékenységeket el kell indítani, vagy engedélyezni kell azok folytatását. Bizonyos idő után (általában 2-3 hét) a folyamat átvizsgálendő és az eltérések feljegyzendőek. Mivel a projekt paramétereinek becslése csak hozzávetőleges, a tervnek mindig szüksége van a módosításokra.

```
Megállapítani a projekt megszorításait
A projekt paramétereinek egy kezdeti összegzését elkészíteni
Definiálni a projekt mérföldköveit és részeredményeit
while a projekt még nincs kész vagy még nem vonták vissza loop
    Felvázolni a projekt ütemtervét
    Elindítani a tevékenységeket az ütemterv alapján
    Várakozni (egy darabig)
    Átvizsgálni a projekt előrehaladását
    Felülvizsgálni a projekt paramétereinek becslését
    Frissíteni a projekt ütemtervét
    Újratárgyalni a projekt megszorításait és részeredményeit
    if (probléma merül fel) then
        Elindítani a technikai felülvizsgálatokat és a lehetséges átdolgozásokat
    end if
end loop
```

5.2. ábra. Projekt tervezése

A projektmenedzsereknek felül kell vizsgálniuk a projekttel kapcsolatos feltételezéseiket, ahogy egyre több információ áll rendelkezésükre. Újra kell tervezniük a projekt ütemtervét. Ha a projekt késik, újra kell tárgyalniuk a megrendelővel a projekt megszorításait és a leszállítandó részeket. Ha ez sikertelen és az ütemterv nem tartható, technikai felülvizsgálatra van szükség. Ennek a felülvizsgálatnak az a célja, hogy valamilyen alternatív megközelítési módot találjon a fejlesztéshez, amely a projekt megszorításaiba még belefér és megfelel az ütemtervnek.

Természetesen a bölcs projektvezető nem feltételezi, hogy minden jól fog menni. Bizonyosfajta problémák csaknem mindig felmerülnek a projekt ideje alatt. A kezdeti feltevéseknek és ütemterveknek inkább pesszimistáknak érdemes lenniük, mint optimistáknak. Érdemes minden eshetőségre, vészhelyzetre felkészülni, így a projekt megszorításait és a mérföldköveket nem szükséges a tervezés minden ciklusában újratárgyalni.

5.2.1. A projektterv

A projektterv felsorolja a projekthez rendelkezésre álló erőforrásokat, a munka felosztását és a munka végrehajtásának ütemtervét is. Néhány szervezetben a projektterv egy egyszerű dokumentum, amely minden egyes különböző típusú tervet (5.1. ábra) magában hordoz. Más esetekben a projektterv kizárólag a fejlesztési folyamattal foglalkozik. Tartalmaz az egyéb tervekre történő hivatkozásokat, de a tervek maguk külön találhatók meg.

Az a tervszerkezet, amelyet itt bemutatok, ez utóbbi típusú tervre vonatkozik. A projektterv részletezései a projekt típusától és a szervezetek típusától függően változnak. Viszont minden egyes tervnek ajánlott tartalmaznia a következő szakaszokat:

1. *Bevezetés.* Ez a projekt céljainak tömör leírása, felsorolja a megszorításokat is (például költségvetés, idő stb.), amelyek befolyásolják a projekt menedzselését.
2. *Projektszervezet.* Megadja a projektcsapat összeállításának módját, a részt vevő embereket és azok szerepét.
3. *Kockázatelemzés.* Leírja a projekt lehetséges kockázatait, azok bekövetkezésének valószínűségét és a kockázat csökkentésének ajánlott stratégiáját. A kockázatkezeléssel az 5.4. alfejezet foglalkozik.
4. *Hardver- és szoftvererőforrás-követelmények.* Azt kell megadni, hogy milyen hardver és milyen támogatott szoftver szükséges a fejlesztéshez. Ha a hardvert meg kell vásárolni, megbecsüli annak árát és a beszerzésének határidejét.
5. *Munka felosztása.* Megadja a projekt tevékenységekre történő felosztását, és azonosítja a tevékenységekhez kapcsolódó mérföldköveket és részeredményeket. Ezekkel az 5.2.2. alfejezet foglalkozik.
6. *Projekt ütemterve.* Meghatározza a tevékenységek közötti függőségeket, megbecsüli a mérföldkövek eléréséhez szükséges időket, és javaslatot tesz a tevékenységekhez rendelendő emberekre.
7. *Figyelési és jelentéskészítési mechanizmusok.* Meghatározza a menedzser által előállítandó jelentéseket, azok előállításának idejét és a használandó projektfigyelési, monitorozási technikát.

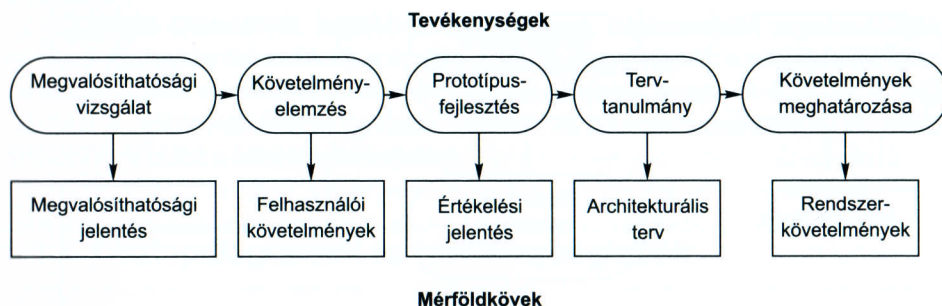
A projekttervet rendszeresen át kell vizsgálni a projekt ideje alatt. Az olyan részek, mint például a projekt ütemezése, rendszeresen meg fognak változni, míg egyéb részek pedig stabilan megmaradnak. Olyan dokumentumszervezést érdemes használni, mely lehetővé teszi az egyszerű cseréket, bővítéseket a dokumentumban.

5.2.2. Mérföldkövek és részeredmények

A menedzsereknek információra van szükségük. A szoftver kézzel nem fogható dolog, az információk csak dokumentum formájában biztosíthatók, amelyek a fejlesztés alatt álló szoftver állapotát írják le. Ezek nélkül az információk nélkül lehetetlenné válik megítélni a haladást, megbecsülni a költségeket és frissíteni az ütemtervet.

Mikor egy projektet tervezünk, egy sor *mérföldkővet* kell meghatároznunk, ahol minden egyes mérföldkő egy szoftverfolyamat-tevékenység végpontja. Minden egyes mérföldkőnél érdemes formális kimenetet készíteni, mint például egy jelentést, amely bemutatható a vezetésnek. A mérföldkőjelentések nem szükségszerűen nagy dokumentumok. Akár a projekt tevékenységei teljesítésének egyszerű rövid leírása is lehet. A mérföldköveket úgy érdemes reprezentálni, mint a projekt logikai szakaszainak végeit. Az olyan határozatlan mérföldköveket, mint például hogy „a kód 80 százaléka kész”, lehetetlen validálni, így a projekt vezetősége számára használhatatlanok.

A *részeredmények* már átadhatók a megrendelőnek. Ezeket általában bizonyos nagyobb projektfázisok, mint például a specifikáció, tervezés stb. végén szállíta-



5.3. ábra. Mérföldkövek a követelményfolyamatban

nak le. A részeredmények, leszállítható résztermékek általában mérföldkövek, de a mérföldkövek nem szükségszerűen részeredmények. A mérföldkövek lehetnek belső projekteredmények, melyeket a projektmenedzser arra használ, hogy ellenőrizze a projekt előrehaladását, de azokat nem adják át a megrendelőnek.

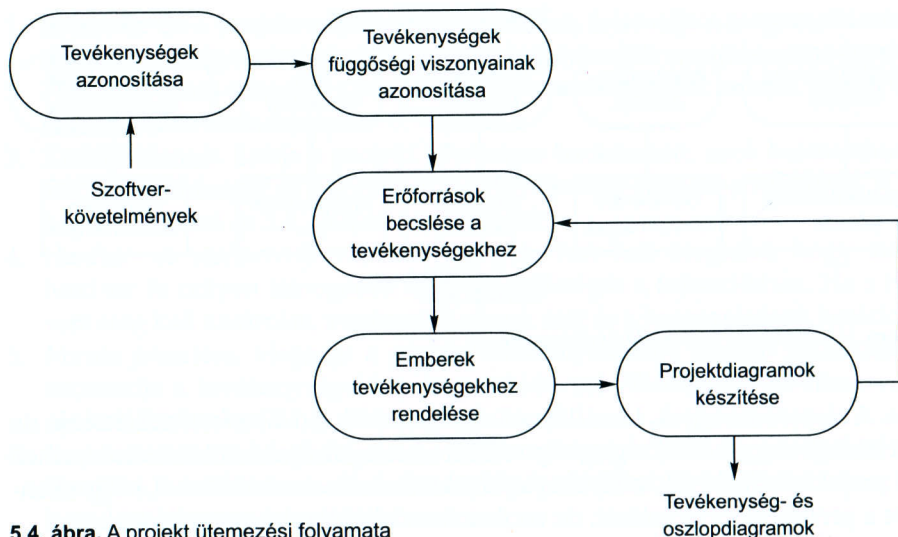
A mérföldkövek meghatározásához a szoftverfolyamatot alapvető tevékenységekre és hozzájuk kapcsolódó kimenetekre kell tördelni. Például az 5.3. ábra egy a követelményspecifikációban érintett tevékenységet mutat be, mikor prototípus-készítést használunk a követelmények validációjához. A tevékenységek alapvető kimeneteit is láthatjuk (a projekt mérföldköveit). A projekt részeredményei a követelmények meghatározása és a követelményspecifikáció.

5.3. A PROJEKT ÜTEMEZÉSE

A projekt ütemezése a projektmenedzserek igen fontos feladata. A menedzserek megbecsülik a tevékenységek elvégzéséhez szükséges időt és erőforrásokat, és egy összefüggő szekvenciába rendezik őket. Hacsak a projekt ütemezése nem egyezik meg teljes mértékben az előző projekttel, annak becslései bizonytalan alapjai az új projekt ütemezésének. Az ütemtervbecslések tovább bonyolódnak azzal, hogy a különböző projektek különböző tervezési módszereket és implementációs nyelveket használhatnak.

Ha a projekt technikailag fejlettebb, a kezdeti becslések majdnem mindig optimistábbak lesznek, még akkor is, ha a menedzserek megpróbálnak minden eshetőséget figyelembe venni. Ebből a szempontból a szoftver ütemezése nem különbözik nagyban egyéb más típusú nagy fejlettségű projekt ütemezésétől. Az újabb légi járművek, hidak és új autómódellek gyakran késnek az előre nem várt problémák felbukkanása miatt. Az ütemterveket ezért folyamatosan frissíteni kell, ahogy egyre több információ áll rendelkezésünkre.

A projekt ütemezése (5.4. ábra) magában foglalja a projekt teljes munkájának különálló tevékenységekre bontását és a tevékenységek elvégzéséhez szükséges idő megítélését is. Általában ezek közül a tevékenységek közül néhány párhuzamosan is elvégezhető. A projekt ütemezőinek kell koordinálniuk ezeket a pár-



5.4. ábra. A projekt ütemezési folyamata

huzamos tevékenységeket, és úgy megszervezniük a munkát, hogy a munkaerő kihasználtsága optimális legyen. El kell kerülniük az olyan szituációkat, ahol a teljes projekt azért csúszik, mert egy kritikus feladat még nincs befejezve.

A projekttevékenységek normális esetben legalább egy hétig tartanak. A finomabb felosztás azt jelenti, hogy aránytalanul sok időt kell szánni a becslésre és a felülvizsgálatra. Szintén hasznos, ha felső időkorlátot szabunk azokra a tevékenységekre, amelyek 8-10 hétig tartanak. Ha ezt túllépik, akkor ajánlatos a tervet és az ütemezést részekre bontani.

Az ütemtervek megállapításakor a menedzsereknek nem ajánlatos azt feltételezniük, hogy a projekt minden egyes szakasza problémamentes lesz. A projekten dolgozó egyének megbetegedhetnek vagy eltávozhatnak, a hardverek meghibásodhatnak, vagy az elengedhetetlen szoftvereket és hardvereket késve szállítják. Ha a projekt még új és technikailag is újszerű, bizonyos részeiről kiderülhetnek, hogy nehezebbek és hosszabb ideig tartanak, mint azt gondoltuk.

A menedzsereknek mindig meg kell becsülni minden egyes tevékenységhez a szükséges erőforrásokat és naptári időt. Az alapvető erőforrások a szükséges emberi erők. Egyéb más erőforrások lehetnek például a szükséges lemezterületek a szerveren, a speciális hardverekre, például szimulátorokra áldozott idő, vagy a projekt munkatársainak utaztatási költségvetése is. Ezek becslését a 26. fejezetben bővebben tárgyalom.

Akkor jó a saccolás ezek megállapítása esetén, ha semmiben sem tévedünk, és a becslésünk kiterjed a nem várt problémákra is. Ez az extra vészhelyzeti faktor a projekt típusától, a folyamat paramétereitől (határidők, szabványok stb.), illetve a projekten dolgozó szoftvertervezők szakértelmének minőségétől is függ. A becsléskor én mindig rászámolok 30 százalékot az eredeti becslésekre a nem várt problémák miatt, és további 20 százalékot az olyan dolgok fedezésére, amelyekre a tervezéskor nem gondoltam.

A projekt ütemtervét legtöbbször diagramok halmazaként reprezentálják, amelyek a munka felosztását, a tevékenységek függőségeit, a munkaerők elhelyezkedését stb. szemléltetik. Ez a következő alfejezet tárgya. A diagramok előállításának automatizálására gyakran használnak szoftvermenedzselési eszközöket, mint például a Microsoft Projectet.

5.3.1. Oszlopdiagramok és tevékenységshálók

Az oszlopdiagramok és a tevékenységshálók grafikus jelölések, melyeket a projekt ütemtervének reprezentálására használunk. Az oszlopdiagramok azt mutatják meg, ki a felelős a tevékenységért, és a tevékenységet mikorra ütemezték be, azaz mikor kezdődik és mikor ér véget. A tevékenységshálók a projektet felépítő különböző tevékenységek közötti függőségeket ábrázolják. Az oszlopdiagramok és a tevékenységshálók a projekt információs adatbázisából projektmenedzselési eszközök segítségével automatikusan generálhatók.

Tekintsük az 5.5. ábra által mutatott tevékenységshalmazt: a tevékenységeket, azok időtartamát és egymástól vett kölcsönös függőségeiket mutatja. Az 5.5. ábrából láthatjuk, hogy a T3 tevékenység függ a T1 tevékenységtől. Ez azt jelenti, hogy a T3 kezdete előtt a T1 tevékenységet be kell fejezni. Például a T1 lehet a komponenstervezés elkészítése, a T3 pedig annak implementációja. Mielőtt az implementáció elindulna, a tervnek kész kell lennie.

Tevékenység	Időtartam (nap)	Függőségek
T1	8	
T2	15	
T3	15	T1 (M1)
T4	10	
T5	10	T2, T4 (M2)
T6	5	T1, T2 (M3)
T7	20	T1 (M1)
T8	25	T4 (M5)
T9	15	T3, T6 (M4)
T10	15	T5, T7 (M7)
T11	7	T9 (M6)
T12	10	T11 (M8)

5.5. ábra. Tevékenységek időtartamai és függőségeik

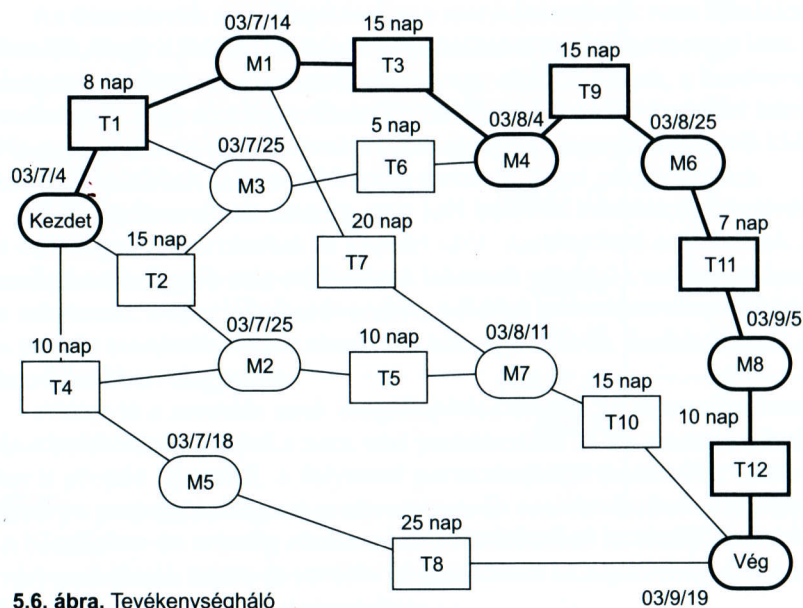
A tevékenységekhez adott függőségek és becsült időtartamuk ismeretében már könnyen generálható a tevékenységsháló, amely a tevékenységek sorrendjét mutatja (5.6. ábra). Azt is észrevehetjük, hogy mely tevékenységeket lehet párhuzamosan végrehajtani, és mit kell egymás után, mert azok korábbi tevékenységektől függnak. A tevékenységeket téglalapokkal, a mérföldköveket és a projekt részeredményeit pedig lekerekített sarkú téglalapokkal jelöltük. A diagramba írt dátumok a tevékenységek kezdetének időpontjai magyar dátumformátum szerint (év/hó/nap). A hálót balról jobbra és fentről lefelé kell olvasni.

Az ilyen diagramok előállításához használt projektmenedzselési eszközökben minden egyes tevékenységet mérföldkövekkel kell lezárni. Egy tevékenység akkor kezdődhet el, mikor az azt megelőző mérföldkövet (mely számos tevékenységtől függhet) elértük. Ezért az 5.5. ábra harmadik oszlopában a megfelelő mérföldköveket (például M5) is feltüntettem, amelyekhez akkor kell elérkezni, amikor a mellettük található tevékenységek ebben az oszlopban befejeződtek (5.6. ábra).

Mielőtt áthaladnánk egyik mérföldköztől a másikhoz, az összes odavezető utat be kell járnunk. Például a T9 tevékenységet, az 5.6. ábrán láthatóan nem kezdhetjük el addig, amíg a T3 és a T6 tevékenységek nincsenek befejezve. Az M4 mérföldkő elérése mutatja meg, hogy ezek a tevékenységek befejeződtek.

A projekt befejezéséhez szükséges legrövidebb idő megbecsléséhez tekintsük a tevékenységsháló leghosszabb útját (a kritikus út). Ebben az esetben ez 11 hét vagy 55 munkanap. Az 5.6. ábrán a kritikus utat a megvastagított tevékenységek sora alkotja. A projekt teljes ütemterve függ a kritikus úttól. Bármelyik kritikus tevékenység teljesítésében bekövetkező időbeli csúszás az egész projekt késéséhez vezet.

Az olyan tevékenységekben bekövetkező késések, melyek nem a kritikus úton vannak, nem szükségképpen okozzák a projekt teljes ütemtervének a csúszását.



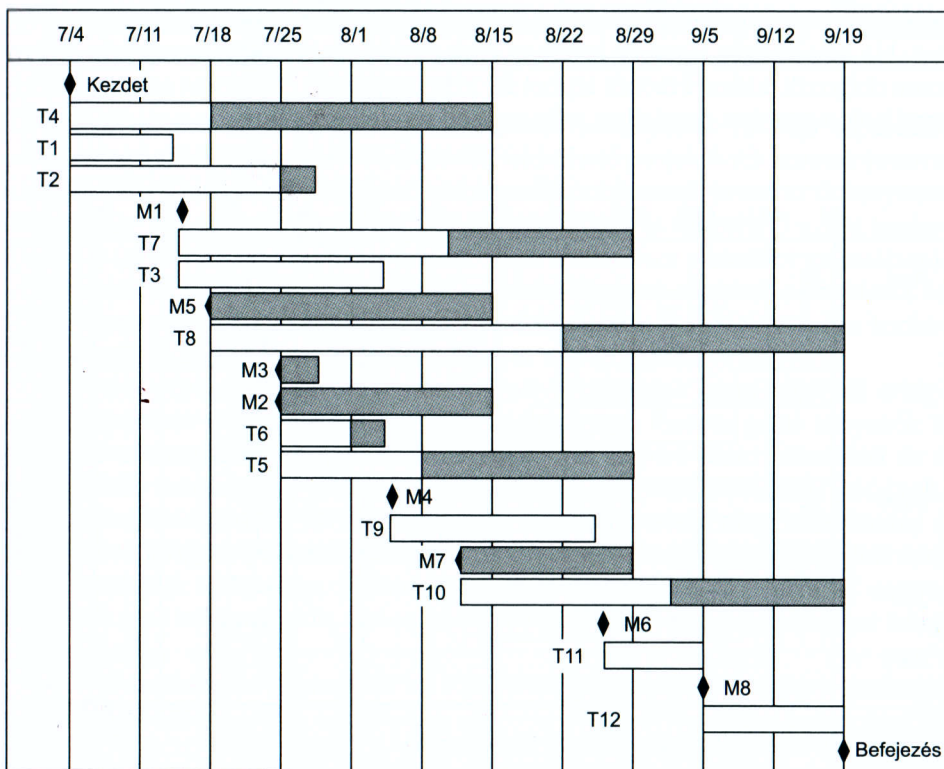
5.6. ábra. Tevékenységsháló

Ha a késések hosszával kiegészítve ezeket a tevékenységeket, azok nem haladják meg a kritikus utat, akkor ezek a csúszások nem lesznek hatással a projekt ütemtervére. Például ha a T8 csúszik, akkor az nincs hatással a projekt végleges teljesítésének dátumára, mert nem a kritikus úton fekszik. A projekt oszlopdiagramja (5.7. ábra) a sötétített oszlopok segítségével a lehetséges késéseket mutatja.

A menedzserek akkor is használnak tevékenység-hálókat, mikor elosztják a projekten belül a munkát. Ezek bepillantást nyújtanak a tevékenységek függőségeibe, melyek ösztönösen nem nyilvánvalóak. Így lehetségessé válik a projektterv módosítása úgy, hogy a kritikus út lerövidüljön. A projekt ütemezése így módon azért csökkenthető, mert redukálható az az idő, amíg a tevékenységek befejezésére várunk.

A projekt kezdeti ütemterve elkerülhetetlenül pontatlanná válik. Ahogy a projekt fejlődik, úgy a becsléseket érdemes összehasonlítani az aktuálisan eltelt idővel. Ez az összehasonlítás aztán az ütemterv felülvizsgálatának alapjaként használható a projekt későbbi szakaszaiban. Mikor az aktuális számadatok ismertetté válnak, a tevékenységdiagramokat is érdemes átvizsgálni. Később a projekttevékenységek újraszervezhetőek, hogy csökkentjük a kritikus út hosszát.

Az 5.7. ábra az ütemezési információk reprezentációjának csak egy lehetséges módja. Ez egy egyszerű oszlopdiagram (*Gantt-diagram*nak is nevezik a bevezetője



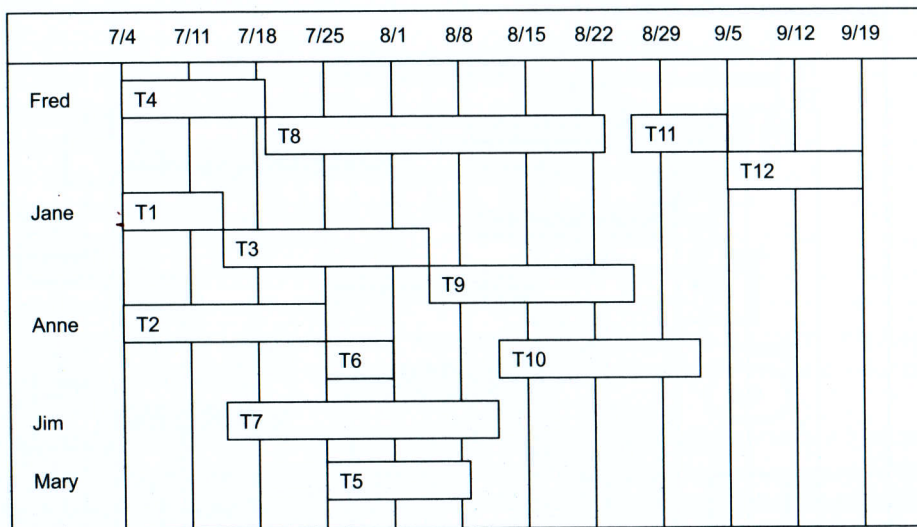
5.7. ábra. Tevékenységdiagram

után), amely a projekt naptárát ábrázolja a tevékenységek kezdeti és befejezési idejével.

Az 5.7. ábrán látható tevékenységekből néhányat egy sötétített oszlop követ, amelynek a hossza ütemezési eszközök segítségével számolható. Ezek azt mutatják, hogy van valamennyi rugalmasság ezen tevékenységek teljesítési idejében. Ha egy tevékenységet nem sikerül időben befejezni, az a kritikus útra nincs hatással addig, amíg a tényleges befejezés ideje a sötétített oszloppal jelzett időn belülre esik. Azoknak a tevékenységeknek, amelyek a kritikus úton helyezkednek el, nem lehet hibájuk, azaz nincs hibahatáruk, így azok könnyen azonosíthatók, hiszen nem követi őket sötétített oszlop.

A projektmenedzsernek nemcsak az ütemtervvel kell foglalkoznia, hanem az erőforrások elosztásával és a munkatársak projekttevékenységekhez történő rendelésével is. Az 5.8. ábra a tevékenységekhez rendel munkatársakat, mely projektmenedzselési eszközök segítségével is előállítható, és így egy oszlopdiaagram generálható, és azt mutatja meg, hogy melyik munkatárs mikor van alkalmazva az adott projektben. A munkatársakat nem szabad a projekt teljes idejében a projekthez rendelni. Periodikusan közbe kell ékelni olyan időszakokat, amikor szabadságra mehetnek, más projekten dolgozhatnak, továbbképzéseken vehetnek részt, vagy más egyéb tevékenységeket végezhetnek.

A nagy szervezetek gyakran alkalmaznak egy sor olyan specialistát, akik a projekten az igényelt szakaszokban dolgoznak. Ez ütemezési problémákat okozhat. Ha egy projekt ugyanis késik az alatt az idő alatt, míg a specialista éppen azon dolgozik, akkor ütközés léphet fel más projektekkel. Azokat szintén késleltetni kell, ugyanis a specialista még nem áll rendelkezésükre.



5.8. ábra. Munkatársak lekötöttsége az idő függvényében

5.4. KOCKÁZATKEZELÉS

A kockázatkezelés egyre növekvő szerepet kap a projektmenedzserek munkájában. Fontos feladatuk a projekt ütemezése közben fellépő, vagy a fejlesztett szoftver minőségében változást előidéző kockázatok megbecslése és azok elkerülése érdekében tett lépések elvégzése (Hall, 1998; Ould, 1999). A kockázatelemzés eredményeit érdemes dokumentálni a projekttervben a kockázat bekövetkezése következményeinek elemzésével egyetemben. A hatékony kockázatkezelés megkönnyíti a problémák kezelését, és lehetővé teszi, hogy az ne emelje meg túlságosan a költségvetést vagy az ütemezett határidőket.

Leegyszerűsítve úgy kell a kockázatra tekinteni, mint annak a valószínűségére, hogy valamilyen ellenséges mozzanatok aktuálisan bekövetkeznek. Kockázatok fenyegethetik a fejlesztett a projektet, a szoftvert és a szervezetet is. Ezek a kockázati kategóriák az alábbi módon definiálhatók:

1. *Projektkockázat.* Olyan kockázat, amely kihatással lehet a projekt ütemtervére vagy az erőforrásokra. Ere példa lehet a tapasztalt tervező hiánya.
2. *Termékkockázat.* Olyan kockázat, amely a fejlesztett szoftver minőségére vagy teljesítményére gyakorol hatást. Ilyen például, ha egy beszerzett komponens nem nyújtja az elvárt teljesítményt.
3. *Üzleti kockázat.* Olyan kockázat, amely a szoftver beszerzését vagy fejlesztését végző szervezetre van hatással. Például ha a konkurens egy új terméket vezet be.

Természetesen ez nem egy egymást kizáró osztályozás. Ha egy tapasztalt programozó elhagyja a projektet, akkor az tekinthető projektkockázatnak (mert a rendszer leszállítása késhet), tekinthető termékkockázatnak (mert az őt helyettesítő esetleg nem annyira tapasztalt, és hibákat vétet), és tekinthető üzleti kockázatnak is (mert ez a tapasztalat már nem áll rendelkezésre a későbbi üzletekben).

A projektre hatással levő kockázatok típusai függnek magától a projekttől és attól a szervezeti környezettől, ahol a szoftvert fejlesztik. Azonban számos kockázat univerzális, ezekből mutatok be néhányat az 5.9. ábrán.

A kockázatkezelés nagyon fontos a szoftverprojektek szemszögéből a legtöbb projektben benne rejlő bizonytalanságok miatt. Ezek a gátló tényezők a pontatlanul meghatározott követelményekből, az egyéni szaktudásokból és a megrendelő szükségletei szerinti követelmények megváltoztatásából adódnak. A projektmenedzszernek fel kell készülnie a kockázatokra, meg kell értenie a kockázatok projektre, termékre és üzletre kifejtett hatását, és lépéseket kell tennie elkerülésük érdekében. Felvázolható egy vészhelyzeti terv arra az esetre, ha a kockázat bekövetkezik, akkor azon nyomban milyen intézkedéseket lehet végrehajtani.

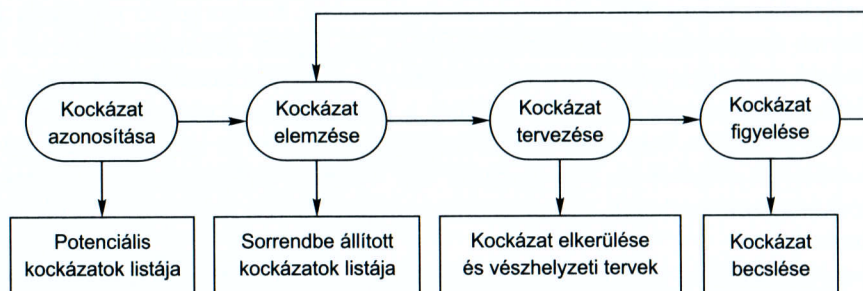
A kockázatkezelés folyamatát az 5.10. ábra illusztrálja. Ez számos szakaszt foglal magában:

1. *Kockázat azonosítása.* A lehetséges projekt-, termék- és üzleti kockázatokat azonosítani kell.

2. *Kockázat elemzése.* A kockázatok valószínűségét és következményeit fel kell becsülni.
3. *Kockázat tervezése.* Terveket kell készíteni a kockázatokhoz azok elkerülése és a projektekre kifejtett hatásuk csökkentése érdekében.
4. *Kockázat figyelése.* A kockázatot folyamatosan fel kell becsülni és a kockázatok enyhítésére adott terveket folyamatosan át kell vizsgálni, ahogy egyre több információ válik elérhetővé a kockázatokról.

Kockázat	Kockázat típusa	Leírás
Munkaerő elpártolása	Projekt	Egy tapasztalt munkaerő elhagyja a projektet annak befejezése előtt.
Vezetőség megváltozása	Projekt	Különböző prioritásokkal megváltozik a szervezeti vezetés.
Hardver elérhetetlensége	Projekt	A projekthez elengedhetetlen hardver nincs időre leszállítva.
Követelmények megváltozása	Projekt és termék	Nagyobb számú változás következik be a követelményekben, mint azt vártuk.
Specifikáció készítése	Projekt és termék	A szükséges interfészek specifikációi időre nem állnak rendelkezésre.
Méret alábecslése	Projekt és termék	A rendszer méretét alábecsültük.
CASE-eszköz alulteljesítése	Termék	A projektet támogató CASE-eszközök nem olyan eredményesek, mint azt várni lehetett.
Technológia megváltozása	Üzleti	Az a technológia, amelyre az egész rendszer épült, korszerűtlenné válik egy másik technológia miatt.
Termékverseny	Üzleti	Egy versenyképes termék kerül piacra, mielőtt a rendszer elkészülne.

5.9. ábra. Lehetséges szoftverkockázatok



5.10. ábra. A kockázatkezelés folyamata

A kockázatkezelési folyamat, a projekttervezéshez hasonlóan, iteratív folyamat, amely a projekt teljes hosszában jelen van. Miután egy kezdeti tervet felvázoltunk, az állapotok megfigyelhetők. Ahogy egyre több információ válik elérhetővé a kockázatokról, úgy azokat újra elemezni kell, és új prioritásokat kell meghatározni. A kockázat elkerülése és a vészhelyzeti terv is módosítható, ahogy újabb kockázati információk kerülnek elő.

A kockázatkezelési folyamat eredményeit is érdemes dokumentálni a kockázatkezelési tervben. Ez magában foglalhatja a kockázat megvitatását a projekt szemszögéből, az ilyen kockázatok elemzését és az ezekre a kockázatokra vonatkozó kezelési tervet. Ahol szükséges, ott néhány kockázatkezelési eredményt is lehet mellékelni, azaz például a kockázat bekövetkezése esetén végrehajtandó specifikus vészhelyzeti terveket.

5.4.1. Kockázat azonosítása

A kockázat azonosítása a kockázatkezelés első szakasza. Ez a projekt lehetséges kockázatainak a felderítését is érinti. Általában a gyakorlatban a kevés következménnyel járó vagy nagyon valószínűtlen kockázatokat nem mindig veszik figyelembe, bár azokat nem kell ebben a szakaszban meghatározni és rangsorolni.

A kockázat azonosítása akár egy csapatmunkaként is végrehajtható egy ötletgyűjtési megközelítési módot alkalmazva, vagy egyszerűen a menedzser tapasztalatain is alapulhat. Hogy segítsük a folyamatot, a lehetséges kockázattípusok listáját is felhasználhatjuk. Ezek a következő típusokat tartalmazhatják:

1. *Technológiai kockázat.* Olyan kockázat, amely a fejlesztés alatt álló rendszer részeként használt szoftver- és hardvertechnológiákból származik.
2. *Emberi kockázat.* Olyan kockázat, amely a fejlesztő csapat tagjaival kapcsolatos.
3. *Szervezeti kockázat.* Olyan kockázat, amely a fejlesztett szoftver szervezeti környezetéből adódik.
4. *Eszközkockázat.* A CASE-eszközökből és az egyéb fejlesztéshez használt szoftveres támogatásokból eredő kockázat.
5. *Követelménykockázat.* A megrendelő követelményeinek megváltozásából és a követelmények kezelésének folyamatváltozásaiból származó kockázat.
6. *Becslési kockázat.* Olyan kockázat, mely a vezetőség a rendszer karakterisztikájára és a rendszer felépítéséhez szükséges erőforrásokra adott becsléséből származik.

Az 5.11. ábra néhány példát ad a lehetséges kockázatokra minden egyes kategóriában. A folyamat végeredménye az esetlegesen felmerülő kockázatok egy hosszú listája lehet, amelyek hatással lehetnek a termékre, a folyamatra vagy az üzletre.

Kockázattípus	Lehetséges kockázat
Technológiai	A rendszerhez használt adatbázis nem tud másodpercenként annyi tranzakciót feldolgozni, mint amennyit elvárunk tőle. Az újrafelhasználható szoftverkomponensek olyan hibákat tartalmaznak, melyek behatárolják azok funkcionalitását.
Emberi	Lehetetlen a megfelelő szakértelemmel rendelkező munkatársakat összehozni. A kulcsfontosságú munkaerő megbetegszik vagy a kritikus időben nem áll rendelkezésre. A szükséges munkaerő-továbbképzés elérhetetlen.
Szervezeti	A szervezetet újrastrukturálják, így más vezetéshez kerül a projekt. A szervezetben lévő pénzügyi problémák a projekt költségvetésének csökkentését erőltetik ki.
Eszköz	A CASE által generált kód nem hatékony. A CASE-eszközöket nem lehet összeintegrálni.
Követelmény	A követelmények szükséges megváltoztatása a terv nagyobb mértékű átdolgozását kívánja meg. A megrendelők nem képesek megérteni a követelmények megváltozásának a hatásait.
Becslési	A szoftver kifejlesztéséhez szükséges időt alábecsülték. A hibajavítási rátát alábecsülték. A szoftver méretét alábecsülték.

5.11. ábra. Kockázatok és kockázattípusok

5.4.2. Kockázat elemzése

A kockázat elemzési folyamata alatt minden egyes azonosított kockázatot sorban át kell tekinteni, és meg kell ítélni annak valószínűségét és komolyságát. Nincs igazán könnyű módja ennek – ez a projektmenedzser tapasztalatától, illetve a megítéléstől függ. Az így kapott eredmények általában nem precíz számszerű értékelések, de számszerű sávokhoz rendelhetők.

1. A kockázat valószínűsége nagyon kicsinek (<10 százalék), kicsinek (10–25 százalék), mérsékeltnek (25–50 százalék), magasnak (50–75 százalék) vagy nagyon magasnak (>75 százalék) tekinthető.
2. A kockázat hatása katasztrofális, súlyos, elviselhető vagy nem jelentős lehet.

Az elemzési folyamat eredményeit érdemes egy táblázatba helyezni, azok súlyossága szerint rendezve őket. Az 5.12. ábra egy ilyen táblázatot mutat be, melyben az 5.11. ábra kockázatait találhatók. Nyilvánvalóan a valószínűségek becslése és a komolyságok megjelölése hasraütésszerűen történt. A gyakorlatban részletesebb információkra van szükségünk a projekttel, a folyamattal, a fejlesztő csapattal és a szervezettel kapcsolatban, hogy a becsléseket megtehessek.

Kockázat	Valószínűség	Hatás
A szervezetben lévő pénzügyi problémák a projekt költségvetésének csökkentését erőltetik ki.	Alacsony	Katasztrofális
Lehetetlen a megfelelő szakértelemmel rendelkező munkatársakat összetoborozni.	Magas	Katasztrofális
A kulcsfontosságú munkaerő megbetegszik vagy a kritikus időben nem áll rendelkezésre.	Mérsékelt	Súlyos
Az újrafelhasználható szoftverkomponensek olyan hibákat tartalmaznak, melyek behatárolják azok funkcionalitását.	Mérsékelt	Súlyos
A követelmények szükséges megváltoztatása a terv nagyobb mértékű átdolgozását kívánja meg.	Mérsékelt	Súlyos
A szervezetet újrastrukturálják, így más vezetéshez kerül a projekt.	Magas	Súlyos
A rendszerhez használt adatbázis nem tud másodpercenként annyi tranzakciót feldolgozni, mint amennyit elvárunk tőle.	Mérsékelt	Súlyos
A szoftver kifejlesztéséhez szükséges időt alábecsülték.	Magas	Súlyos
A CASE-eszközöket nem lehet összeintegrálni.	Magas	Elviselhető
A megrendelők nem képesek megérteni a követelmények megváltozásának a hatásait.	Mérsékelt	Elviselhető
A szükséges munkaerő-továbbképzés elérhetetlen.	Mérsékelt	Elviselhető
A hibajavítási rátát alábecsülték.	Mérsékelt	Elviselhető
A szoftver méretét alábecsülték.	Magas	Elviselhető
A CASE által generált kód nem hatékony.	Mérsékelt	Nem jelentős

5.12. ábra. Kockázatok és kockázatelemzés

Természetesen mind a kockázat valószínűsége, mind a kockázat hatása megváltozhat, mihelyt több információ válik a kockázatról elérhetővé, és a kockázatkezelési tervek implementálásra kerülnek. Így ezt a táblázatot is frissíteni kell a kockázati folyamat minden egyes iterációjában.

Amint egy kockázatot kielemeztünk és rangsoroltunk, meg kell ítélni, melyik a legfontosabb kockázat, amelyet a projekt ideje alatt végig figyelemmel kell kísérni. Ez a megítélés függ a felmerülő kockázati valószínűségek kombinációjától és azok hatásaitól is. Általánosan elmondható, hogy érdemes minden olyan katasztrofális kockázatot és súlyos kockázatokat figyelemmel kísérni, melyek bekövetkezésének valószínűsége nagyobb, mint mérsékelt.

Boehm (Boehm, 1988) azt javasolta, hogy azonosítsuk és figyeljük meg a „top 10” kockázatot, de én úgy gondolom, hogy ez a kijelentés nagyon önkényes. A megfigyelendő kockázatok száma függ a projekttől is. Lehet akár 5, de lehet akár 15 is. Mindazonáltal a megfigyelendő kockázatok száma menedzselhető.

A kockázatok túl nagy száma egyszerűen túl sok begyűjtendő információt követel meg. Az 5.12. ábrán azonosított kockázatok esetében érdemes azt a nyolc kockázatot tekinteni, melyek következményei katasztrofálisak, illetve súlyosak.

5.4.3. Kockázat tervezése

A kockázat tervezési folyamata a kulcskockázatok mindegyikét figyelembe veszi, és meghatározza azok kezelési stratégiáját. Itt ismételtelen nem létezik olyan egyszerű folyamat, melyet követve a kezelési terv megadható. Ez szintén a megítéléséktől és a projektmenedzser tapasztalatától függ. Az 5.13. ábra az 5.12. ábrán azonosított kulcsfontosságú kockázatokra mutat néhány lehetséges stratégiát.

Ezek a stratégiák három nagy kategóriába esnek, ezek:

1. *Elkerülési stratégiák.* Ezeket a stratégiákat követve a kockázatok felmerülésének a valószínűsége csökkenthető. Ilyen kockázatelkerülési stratégiára példa az 5.13. ábrán található hibás komponensekkel foglalkozó stratégia.
2. *Minimalizációs stratégiák.* Ezeket a stratégiákat követve a kockázat hatása csökkenthető. Ilyenre példa az 5.13. ábrában a munkaerő megbetegedésére adott stratégia.
3. *Vészhelyzeti tervek.* Ezeket követve, ha bekövetkezik a legrosszabb, mi már felkészültünk rá, és van helyben ezzel foglalkozó stratégiánk. Ilyen vészhelyzeti tervre példa az 5.13. ábrán a szervezeti pénzügyi problémával foglalkozó stratégia.

Kockázat	Stratégia
Szervezeti pénzügyi problémák	Elkészíteni egy részletes dokumentumot a legfelső vezetés részére, mely bemutatja, hogy a projekt mennyire fontos kapcsolatban áll az üzleti célokkal.
Toborzási problémák	Riasztani a megrendelőt, hogy potenciális nehézségek és lehetséges késések várhatók, kutatni megvásárolható komponensek után.
Munkaerő megbetegedése	Újraszervezni a csapatot úgy, hogy a munkák jobban átfedjék egymást, így az emberek jobban megértik mások munkáit is.
Hibás komponensek	Kicserélni a potenciális hibás komponenseket ismert megbízhatósággal rendelkező megvásárolt komponensekre.
Követelmények megváltozása	Kimutatható információkat szerezni a követelmények megváltoztatása hatásának megállapítására, maximalizálni a tervben rejtőző információkat.
Szervezet újrastrukturálása	Elkészíteni egy részletes dokumentumot a legfelső vezetés részére, mely bemutatja, hogy a projekt mennyire fontos kapcsolatban áll az üzleti célokkal.
Adatbázis-teljesítmény	Kideríteni nagy teljesítményű adatbázisok megvásárolhatóságát.
Alábecsült fejlesztési idő	Megvásárolható komponenseket keresni, programgenerátor használatát megfontolni.

5.13. ábra. Kockázatkezelési stratégiák

Analógiát figyelhetünk meg az itt olvasottak és a kritikus rendszerek megbízhatósága, biztonságossága és üzembiztonsága terén. Fontos kiemelni, hogy mindig érdemes valamilyen stratégiát választani a kockázatok elkerülésére. Amennyiben ez nem lehetséges, törekedni kell a kockázatok hatásának minimalizálására. Végezetül, a megfelelő időben alkalmazott stratégiák a projekt, illetve a termék teljes egészére hatással lesznek.

5.4.4. Kockázat figyelése

A kockázat figyelése magában foglalja annak rendszeres meghatározását, hogy minden egyes azonosított kockázat bekövetkezésének valószínűsége nagyobb vagy kevesebb lett-e, és hogy annak hatása változott-e. Természetesen ez nem mindig figyelhető meg közvetlenül, így egyéb tényezőket is meg kell vizsgálnunk, amelyek adalékokkal szolgálhatnak az adott kockázat valószínűségére és hatására. Ezek a tényezők nyilvánvalóan a kockázattípusok függvényei. Az 5.14. ábra az ilyen kockázattípusok meghatározásában segítséget nyújtó tényezőkre ad példát.

A kockázatfigyelést folytonos tevékenységként érdemes művelni, és minden egyes vezetőségi felülvizsgálat esetén minden kulcskockázatot külön figyelembe kell venni, és azt a találkozón megbeszélni.

Kockázattípus	Potenciális jelzés
Technológiai	A hardver vagy a támogató szoftver kései leszállítása, számos jelzett technológiai probléma.
Emberi	Szegényes munkamorál, rossz kapcsolatok a csapattagok között, munka rendelkezésre állása.
Szervezeti	Munkahelyi pletykák, a legfelső vezetés tevékenységének hiánya.
Eszköz	A csapattagok eszközökkel szembeni idegenkedése, a CASE-eszközök körüli panaszok, nagyobb teljesítményű munkaállomások iránti igény.
Követelmény	Igény számos követelmény megváltoztatására, panaszok a megrendelő felől.
Becslési	Az elfogadott ütemtervet nem sikerül betartani, nem lehet tisztázni a jelentett hibákat.

5.14. ábra. Kockázati tényezők

Kulcsfogalmak

- A szoftverprojekt menedzselése elengedhetetlen, ha a szoftvertervezési projektet időre és adott költségvetésen belül szeretnénk végrehajtani.
- A szoftvermenedzsment különbözik minden más mérnöki menedzsmenttől. A szoftver kézzel nem fogható. A projektek lehetnek újak, illetve innovatívak, így még senkinek sincs menedzselésükkel kapcsolatos tapasztalata. A szoftverfolyamatokat nem mindig értik meg jól.
- A szoftvermenedzsereknek sok szerepük van. A legfontosabb tevékenységük a projekt tervezése, a becslés és az ütemezés. A tervezés, becslés iteratív folyamatok. A projekt teljes ideje alatt folytonosan végighúzódnak tevékenységek. Ahogy egyre több információ áll rendelkezésre, a tervet és az ütemtervet úgy kell folyamatosan felülvizsgálni.
- A projekt mérföldkövei egy tevékenység előre jelezhető kimenetei, ahol a haladásról bizonyos formális jelentések készíthetők a menedzsment felé. Mérföldköveket egy szoftverprojekt folyamán rendszeresen el kell érni. A részeredmények, közbeni termékek olyan mérföldkövek, melyek a projekt megrendelőjének leszállíthatók.
- A projekt ütemezése magában foglalja a különféle grafikus reprezentációk elkészítését mint a projektterv részét. Ezek tevékenységdiagramokat tartalmazhatnak, amelyek a projekt tevékenységei közötti kapcsolatokat mutathatják, és oszlopdiagramokat, amelyek a tevékenységek időtartamait tartalmazzák.
- A nagyobb projektkockázatokat azonosítani kell, és meg kell határozni azok valószínűségeit és következményeit a projektre nézve. A valószínű és potenciálisan súlyos kockázatok elkerülésére, kezelésére terveket kell készíteni, amelyek azokkal a problémákkal foglalkoznak, melyek felmerülhetnek. A kockázatokat meg kell tárgyalni minden egyes projekttalálkozáson.

További irodalom

Waltzing with Bears: Managing Risk on Software Projects. Nagyon praktikus, könnyen olvasható bevezetés a kockázatkezelésbe. (DeMarco, T. és Lister, T., 2003, Dorset House.)

Managing Software Quality and Business Risk. Ennek a könyvnek a 3. fejezete a kockázatok legjobb tárgyalása, amit valaha is olvastam. A könyv a kockázat köré szerveződik, és személyes véleményem szerint ebben a témában az írása idején nagy valószínűséggel ez volt a legjobb könyv. (Ould, M., 1999, John Wiley & Sons.)

The Mythical Man Month (Anniversary Edition). A szoftvermenedzselés problémái nem változtak az 1960-as évektől, és ez az egyik legjobb könyv ebben a témában. Érdekes és jól olvasható számvetése ez az egyik első nagy szoftverprojektnek, az IBM OS/360 operációs rendszernek. A jubileumi kiadás (ami 20 évvel az első, 1975-ös kiadás után jelent meg) Brooks néhány egyéb klasszikus írását is tartalmazza. (Brooks, F. P., 1995, Addison-Wesley.)

Software Project Survival Guide. A szoftvermenedzsment eléggé pragmatikus szemléletét adja, de jól használható gyakorlati példákat tartalmaz. Könnyen olvasható és érthető. (McConnell, S., 1998, Microsoft Press.)

Lásd még a 6. rész menedzseléssel kapcsolatos irodalmait.

Feladatok

- 5.1. A szoftver nem kézzelfogható. Fejtse ki, miért vet fel ez speciális problémákat a szoftverprojekt menedzselésében.
- 5.2. Fejtse ki, miért van az, hogy a legjobb programozók nem mindig a legjobb szoftvermenedzserek. Segítségére lehet, ha válasza alapjául az 5.1. alfejezetben megadott menedzselési tevékenységeket tekinti.
- 5.3. Fejtse ki, hogy a projekt tervezési folyamata miért iteratív, és miért kell a tervet folyamatosan átvizsgálni a szoftverprojekt ideje alatt.
- 5.4. Tömören fejtse ki a szoftverprojekt terv minden egyes szakaszának a céljait.
- 5.5. Mi a kritikus különbség a mérföldkő és a részeredmény között?
- 5.6. Az 5.15. ábra felsorol egy sor tevékenységet, időtartamot és függőségeket. Rajzoljon fel egy tevékenységdiagramot és egy oszlopdiagramot, melyet a projekt ütemtervének bemutatására használna.

Tevékenység	Időtartam (nap)	Függőségek
T1	10	
T2	15	T1
T3	10	T1, T2
T4	20	
T5	10	
T6	15	T3, T4
T7	20	T3
T8	35	T7
T9	15	T6
T10	5	T5, T9
T11	10	T9
T12	20	T10
T13	35	T3, T4
T14	10	T8, T9
T15	20	T12, T14
T16	10	T15

5.15. ábra. Tevékenységek időtartama és és függőségei

- 5.7. Az 5.5. ábra a tevékenységek időtartamát mutatja be szoftverprojekt-tevékenységekhez. Tegyük fel, hogy egy súlyos, nem várt balszerencsés esemény következik be, és a T5 tevékenység 10 nap helyett 40 napig tart. Vizsgálja felül a tevékenység-hálót ennek megfelelően, kiemelve az új kritikus utat. Rajzoljon fel egy új oszlopdiagramot, amely azt mutatja meg, hogyan lehet a projektet újraszervezni.
- 5.8. Az irodalomban megtalálható projektprobléma példák segítségével sorolja fel a menedzselés nehézségeit, amelyek azokban a félresikerült programozói projektben felmerültek. (Kezdje Brooks könyvével!)
- 5.9. Az 5.11. ábrán mutatott kockázatokon kívül adjon meg hat további lehetséges kockázatot, amelyeknek megjelenése várható egy szoftverprojektben.
- 5.10. A rögzített árú szerződések, ahol a szerződők előre rögzítik a teljes rendszer kifejlesztésének árát, átmozgatják a projekt kockázatát a klientsztől a szerződéskötőhöz. Ha bármi elromlik, a szerződést kötőnek kell fizetnie. Váolja fel, miért növelhetik az ilyen típusú szerződések a kockázat felmerülésének valószínűségét.
- 5.11. Megkéri önt a menedzser, hogy szállítson le egy szoftvert időre, melyet csak úgy tud teljesíteni, hogy megkéri a projektcsapatát, hogy fizetetlenül is túlórázzanak. A projektcsapat minden egyes tagjának fiatal gyerekei vannak. Fejtse ki, hogy minden további nélkül elfogadná-e ezt a követelést az ön projektmenedzserétől, vagy inkább megbeszelné a projektcsapat tagjaival, hogy hajlandóak lennének-e több időt szentelni a munkának és kevesebbet a családnak? Milyen tényezők játszhatnak fontos szerepet az ön döntésében?
- 5.12. Ön programozó, és felajánlják azt, hogy előléptetik projektmenedzserre, de ön úgy érzi, hogy sokkal hatékonyabban tud közreműködni inkább technikai, mint menedzseri szerepben. Gondolja meg, elfogadná-e az előléptetést.

KÖVETELMÉNYEK

A nagy és komplex szoftverrendszerek fejlesztésénél talán a követelménytervezés a legbonyolultabb probléma.

A követelménytervezés annak megállapítása, hogy mit kellene csinálni a rendszernek, mik a megkívánt és eredendő tulajdonságai, és mik a körülmények a rendszer működésének és a szoftverfejlesztési folyamatnak. Úgy gondolhatunk a követelménytervezésre, mint a szoftver ügyfelei és felhasználói, valamint a szoftver fejlesztői közötti kommunikációra.

A követelménytervezés nem egyszerűen technikai feladat. A rendszerkövetelményeket befolyásolják a felhasználói rokonszenvek és ellenszenvek, az előítéletek, politikai és szervezeti tényezők. Ezek alapvető emberi jellemzők, és az új technológiák (mint például a használati esetek, forgatókönyvek, formális módszerek) sem segítenek megoldani ezt a kényes problémát.

Ezen rész fejezetei két kategóriába esnek – a 6. és 7. fejezetben bevezetem a követelménytervezés alapfogalmait, a 8–10. fejezet a követelménytervezési folyamatban használt modelleket és technikákat ismerteti. Részletesebben:

1. A 6. fejezet témaköre a követelménytervezés és követelmények dokumentálása. Tárgyalom a követelmény fogalmát, a követelmények különböző típusait, és azt, hogy ezeket a követelményeket hogyan lehet egyetlen követelményspecifikáció-dokumentumba szervezni. Ebben a fejezetben bevezetem a második esettanulmányt, a könyvtári rendszert.
2. A 7. fejezetben a követelménytervezési folyamat tevékenységeire fókuszálok. Tárgyalom a megvalósítási tanulmányt (amelynek mindig a követelménytervezés részének kellene lennie), a követelmények feltárását és elemzését és a követelményvalidációt.
3. A 8. fejezet foglalkozik a követelménytervezési folyamatokban alkalmazható rendszermodellekkel. Ezek jóval részletesebb leírási eszközt adnak a rendszerfejlesztők kezébe. A hangsúly az objektumorientált modellezésen van, de foglalkozom az adatfolyam-diagramokkal is. Ezeket intuitívnek és hasznosnak tartom, mert átfogó képet adnak arról, hogyan dolgozza fel a rendszer az információt.
4. A 9. és 10. fejezetben a hangsúly a kritikus rendszerek specifikációján van. A 9. fejezetben az üzembiztonsági jellemzők specifikálását tárgyalom. Leírom a kockázatvezérelt közelítést, valamint a biztonságosság, a megbízhatóság és a védettség specifikációjának speciális elemeit. A 10. fejezetben a formális specifikációs technikákról szólok. A formális módszerek jelentősége kisebb annál, mint amilyenek tartották, de egyre inkább használják őket a biztonságos és küldetéskritikus rendszerek specifikációjánál. Ebben a fejezetben áttekintem az algebrai és modellalapú közelítéseket.