

اندازه گیری نرم افزار

اندازه گیری :

- فرایندی است که در آن اعداد، علائم به خصوصیات اشیاء، براساس قواعد تعریف شده، مشخص میشود و انتصاب مییابد. مثال : وزن صندلی : ۴ kg
- اندازه: یک نشانه کمی از مقدار، میزان، بعد، ظرفیت یا اندازه یک خصوصی از یک محصول یا فرایند مثال : ۲ تا باگ وجود دارد → ۱۰۰۰ خط برنامه
- معیار: یک رابطه کمی بین اندازه ها، که با توجه به نیاز و شرایط مسئله تعریف میشود. مثال : معیار کیفیت کار انجام شده = تعداد متوسط باگ در ۱۰۰۰ خط / یک ماه
- شاخص : یک معیار یا مجموعه ای از معیارهاست که بینشی را نسبت به پروژه، فرایند یا محصول ایجاد میکند.
- مثال : برنامه نویس خوب = $\frac{5000 \text{ line code}}{\text{month}}$

کیفیت توصیف نیازمندی ها

N_{user} برابر است با تعداد نیازمندی هایی که user ها مختلف نظر سازگاری در مورد آنها داشته اند.

مشخص میکند که توصیف نیازمندی آن قدر خوب بوده که از دید افراد مختلف این نیازمندی دید یکسانی را ایجاد کرده است.

$$N_R = N_F + N_{NF}$$

• معیار کیفیت توصیف نیازمندی ها :

$$Q = \frac{N_{user}}{N_R}$$

تعیین اندازه نرم افزار : روش فیزیکی

شمارش Line of Codes

$$Q = \frac{LOC}{month}$$

یعنی تعداد خطوط برنامه در یک ماه به عنوان معیار انتخاب میشود ولی این معیار مناسبی نیست و مشکلاتی دارد:

- وابسته به برنامه نویس
- وابسته به زبان برنامه نویسی
- پیچیدگی الگوریتم ها را در نظر نمیگیرد
- وجود Code generator ها

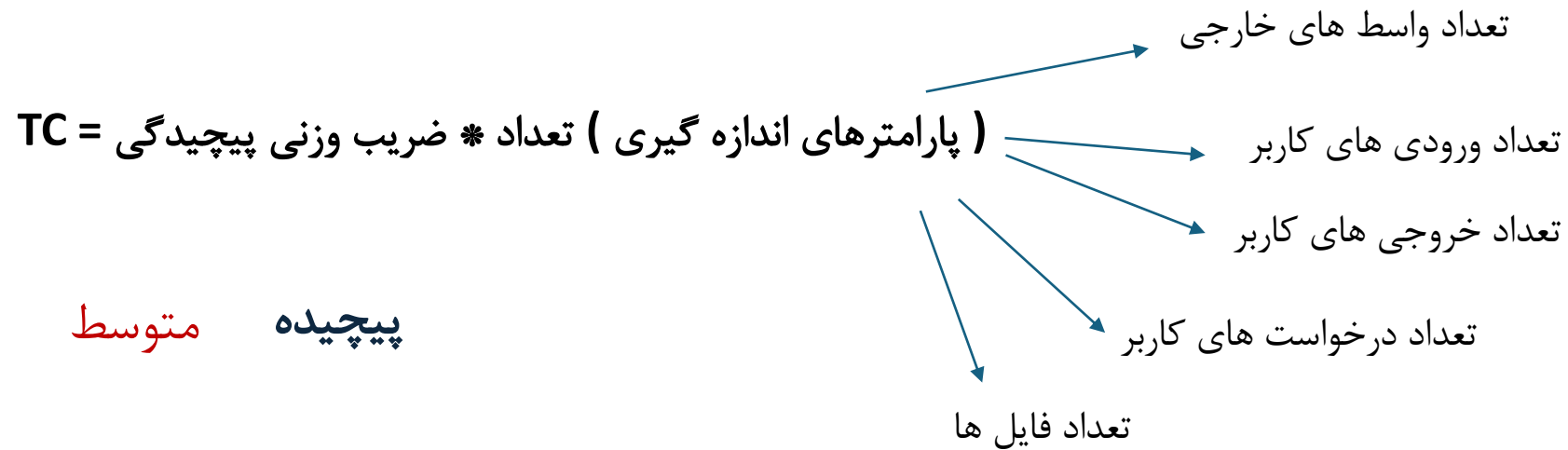
در صورتی که بخواهیم از LOC استفاده کنیم، باید logical loc را در نظر بگیریم (مثلا تعداد خطوط برابر است با تعداد ; ها) ولی در physical loc تعداد خطوط فاصله و کامنت ها نیز شمرده میشوند.

Language Gearing factor

مشخص میکند چند خط کد در زبانهای برنامه نویسی معادل هم هستند

زبان برنامه نویسی	میانگین LOC/FP
اسمبلی	۳۲۰
C	۱۲۸
C++	۶۴
کوپول	۱۰۶
فرترن	۹۰
پاسکال	۹۵
ادا	۵۳
ویژوال بیسیک	۳۲

معیارهای سنجش مبتنی بر عملکرد function-point



پیچیده متوسط ساده

پیچیده	متوسط	ساده	شمارش	پارامترهای اندازه گیری
۶	۴	۳		تعداد ورودیهای کاربر
۷	۵	۴		تعداد خروجیهای کاربر
۶	۴	۳		تعداد درخواستهای کاربر
۱۵	۱۰	۷		تعداد فایلها
۱۰	۷	۵		تعداد واسطهای خارجی

شمارش کل :

$$FP = TC * [0.65 + 0.01 * \sum F_i]$$

F_i : میزان تعدیل پیچیدگی براساس ۱۴ سوال

۱۴ سوال

- ☐ آیا سیستم به پشتیبانی و بازیابی قابل اطمینان نیاز دارد؟
- ☐ آیا ارتباطات داده ای مورد نیاز است؟
- ☐ آیا عملیات پردازشی توزیع شده وجود دارد؟
- ☐ آیا کارایی اهمیت دارد؟
- ☐ آیا سیستم در یک محیط عملیاتی موجود و پرکار اجرا می شود؟
- ☐ آیا سیستم به در خواستهای On-Line نیاز دارد؟
- ☐ آیا در خواستهای On-Line نیاز به ساخت عملیات ورودی روی عملیات یا صفحات نمایش چندگانه دارد؟

☐ آیا فایل‌های اصلی بصورت On-Line به‌نگام می‌شوند؟

☐ آیا ورودی‌ها، خروجی‌ها و فایل‌ها پیچیده‌اند؟

☐ آیا پردازش داخلی پیچیده است؟

☐ آیا کد طوری طراحی شده است که دوباره قابل استفاده باشد؟

۱. تبدیل و نصب در طراحی لحاظ شده است؟

۲. آیا سیستم برای نصب چند گانه در سازمانهای متفاوت طراحی شده است؟

۳. آیا برنامه کاربردی طوری طراحی شده است که تغییرات راتسهیل کند و به آسانی توسط کاربر

استفاده شود؟

0	1	2	3	4	5
بی تاثیر	ضعیف	معقول	متوسط	با اهمیت	اساسی

Use case point

در UCP مراحل زیر باید اجرا شود:

1. تعیین نوع و تعداد Uc ها : UUCP

2. Technical complexity factor (مسائل فنی)

$$TC = 0.6 + 0.01 * \sum F_i$$

F_i : میزان تعدیل پیچیدگی براساس ۱۴ سوال

۳. Environment factor (EF) (جدول محیط)

$$EF = 1.4 + (-0.03 * EFactor)$$

$$UCP = UUCP * TCF * EF \quad ۴.$$

تخمین نرم افزار

شامل موارد زیر می باشد:

- زمان

- هزینه

تخمین هزینه

۱. روش قیاسی :

براساس تجارب پروژه های مشابه برای پروژه کنونی تخمین زده میشود. این روش مناسبی نیست زیرا شرایط محیطی ثابت نیستند.

۲. روش های تجربی :

با استفاده از نظر افراد خبره هزینه های پیشنهاد میشود و با روش های مختلف آن ها را جمع بندی میکنیم.

۳. روش بالا به پایین :

برای هر app یم مقیاس عددی مانند FP را بدست آورده و براساس جداول موجود متناظر اندازه بدست آورده را در نظر میگیریم.

۴. روش پایین به بالا :

نرم افزار را شکسته و زیر فعالیت هایی را بدست می اوریم و برای آنها هزینه را تعیین میکنیم.

این روش مشکلاتی را دارد.

همه فعالیت ها از ابتدا مشخص نیست.

فعالیت های چتری دیده نمیشوند. یعنی فعالیت هایی که در کل طول مسئله باید دیده شوند.

دقت در تخمین هر task

استفاده از مدل‌های تجربی برای برآورد هزینه نرم افزار

مدل های کوکومو (COCOMO)

$$E = A + B * (ev)^c \quad \bullet$$

• مدل COCOMOII

برآورد پروژه های نرم افزاری و روش های آن



- برآورد ساخت یا خرید
- ایجاد درخت تصمیم

$$\text{expected cost} = \sum (\text{path probability}) * \text{estimated path cost}$$

ریسک

ریسک : رویدادی است که ممکن است در آینده رخ دهد و باعث اختلال و تاثیرات منفی در روند پروژه شود.

دو ویژگی کلیدی ریسک:

- **عدم قطعیت** : ریسک ممکن است اتفاق بیفتد یا نه
- **ضایعات** : اگر ریسک به وقوع بپیوندد حتما ضایعاتی در پی خواهد داشت.

رویکردها در مقابله با ریسک

- واکنش (reactive) : پس از وقوع ریسک به رفع اثرات می پردازد.
- پیش واکنش (proactive) : نگاه در پیشگیری دارد.

در رویکرد واکنشی در همان لحظه وقوع تصمیم می گیریم در حالی که در پیش واکنش از قبل تصمیم گیری می شود.

مراحل انجام کار

1. شناسایی ریسک

2. الویت بندی

3. تهیه برنامه مدیریت ریسک Risk management plan

• شناسایی ریسک :

ریسک به دو نوع است :

- ریسک کلی : قطعی برق یا اینترنت
- ریسک خاص یک پروژه : کم شدن نیروهای انسانی

نکات در مواجهه با ریسک

1. اجتناب از ریسک : طوری برنامه ریزی می کنیم که تا حد امکان ریسک ایجاد نشود و تمهیداتی را در نظر بگیریم که مشکلی ایجاد نشود. ولی باید در نظر داشته باشیم که تمهیدات خود منجر به مشکل نشوند.
2. نظارت بر ریسک: بررسی ریسک در طول انجام پروژه ، بررسی اینکه آیا اقدامات پیشگیرانه موثر است ؟ - تکمیل اطلاعات در مورد ریسک
3. مدیریت ریسک و برنامه ریزی احتیاطی : در صورتی که ریسک اتفاق بیفتد، تمهیدات مربوطه را اعمال می کند. مشخص کردن اینکه در صورت بروز ریسک چه تمهیداتی صورت گیرد.

تأثیر ریسک Risk Impact

- فاجعه بار catastrophic
- بحرانی Critical
- حاشیه ای marginal
- قابل چشم پوشی Negligible

گروه های متفاوت ریسک :

- ریسک پروژه ای : زمان - هزینه - نیروی انسانی
- ریسک فنی : کیفیت نرم افزار
- ریسک ها تجاری (بیرونی) : ریسک بازار - قوانین - عوامل اقتصادی

احتمال وقوع ریسک :

- زیاد high
- متوسط moderate
- کم low

جدول ریسک

ریسک‌ها	گروه	احتمال	تاثیر
برآورد اندازه‌ها ممکن است بسیار کم باشد.	PS	۶۰٪	۲
تعداد کاربران بیشتر از حد برنامه‌ریزی شده	PS	۳۰٪	۳
استفاده مجدد کمتر از حد برنامه‌ریزی شده	PS	۷۰٪	۲
مقاومت کاربران نهایی در برابر سیستم	BU	۴۰٪	۳
کم بودن مهلت تحویل	BU	۵۰٪	۲
از دست رفتن سرمایه	CU	۴۰٪	۱
تغییر خواسته‌های مشتری	PS	۸۰٪	۲
عدم برآوردن انتظارات توسط فناوری	TE	۳۰٪	۱
عدم آموزش ابزارها به افراد	DE	۸۰٪	۳
بی‌تجربگی پرسنل	ST	۳۰٪	۲
بالا بودن بازدهی پرسنل	ST	۶۰٪	۲

در این جدول:

PS: ریسک اندازه پروژه

BU: ریسک تجاری

CU: ریسک مشتری

TE: ریسک فناوری

DE: ریسک محیط توسعه

ST: ریسک پرسنلی

ارزش تاثیرات:

۱: فاجعه‌بار

۲: بحرانی

۳: کم‌اهمیت

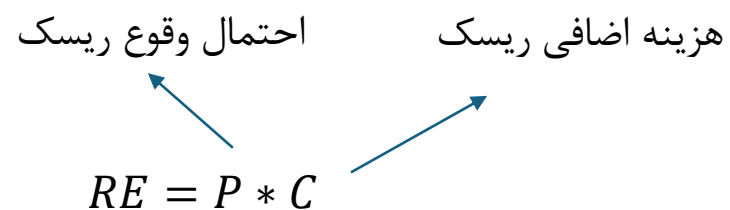
۴: قابل چشم‌پوشی

برآورد ریسک

- قرار گردفتن در معرض ریسک

احتمال وقوع ریسک

هزینه اضافی ریسک

$$RE = P * C$$


- هزینه اضافی ریسک

مشخص می کند در صورتی که ریسک اتفاق بیفتد چقدر خسارت ایجاد می کند.

هزینه LOC هر مولفه * تعداد خطوط هر مولفه * تعداد مولفه های اضافی = C

میزان مصرف هزینه برای ریسک

- براساس قانون ۸۰-۲۰، ۸۰٪ هزینه ها و خسارت های پروژه مربوط به اشکالاتی است که در ۲۰٪ پروژه است، بوقوع پیوسته است. در نتیجه باید ریسک های مهم را شناسایی و کنترل کرد.
- ریسک جزو فعالیت های چتری است و در طول انجام پروژه همیشه باید در نظر گرفته شود.

مدیریت زمان پروژه



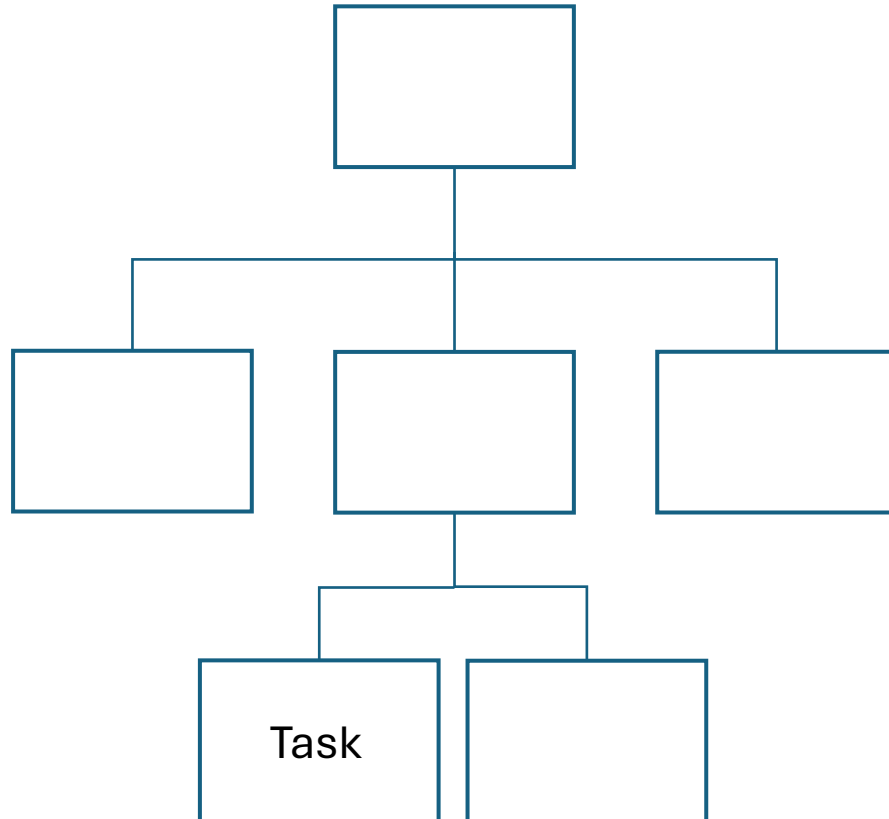
- سه عامل مهم در یک پروژه وجود دارد که باید به تعادل برسیم.

مسائل مطرح در زمانبندی

- شکستن مسئله و تعیین Task ها
- مشخص کردن وابستگی بین Task ها
- تخصیص زمان به هر Task (تاریخ شروع و پایان)
- اعتبارسنجی Task ها
- انتساب وظایف به اعضای تیم
- تعیین دستاوردها

تعیین لیست فعالیت‌ها

در این روش یک پروژه را آنقدر میشکنیم تا به فعالیت‌های تک منظوره برسیم.



تعیین وابستگی بین taskها

انواع وابستگی ها عبارتند از :

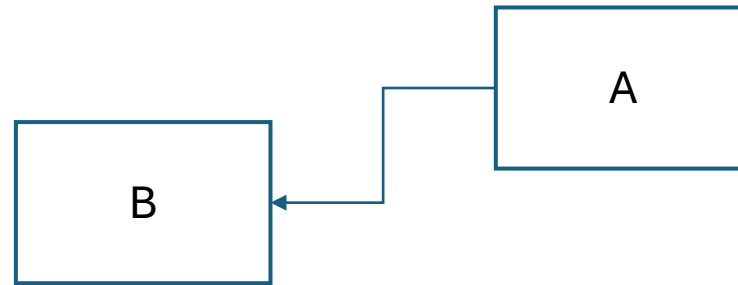
- وابستگی اجباری : جزء ذات کار است.
- وابستگی احتیاطی : توسط تیم تعیین می شود.
- وابستگی خارجی : از خارج از پروژه اعمال می شود. مثلا یک نرم افزار که از خارج از شرکت خریداری می شود.

انواع وابستگی بین taskها

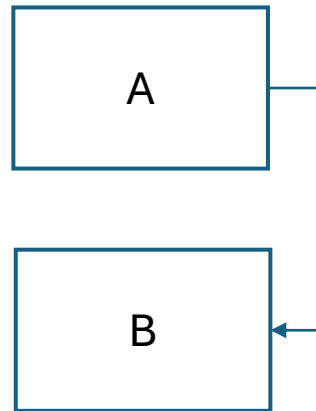
- Finish to start



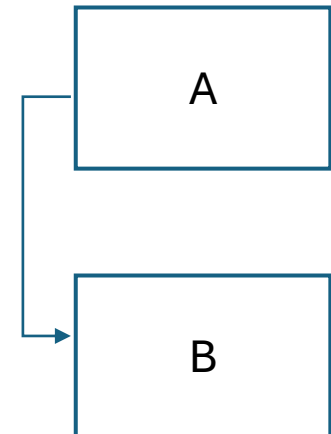
- Start to Finish



- Finish to finish

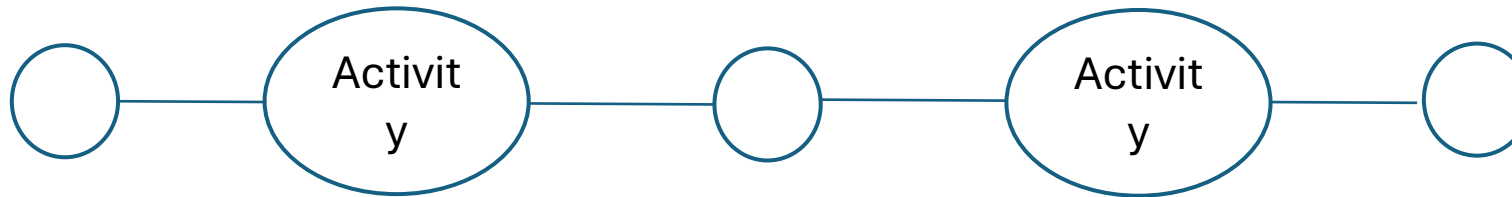


Start to start

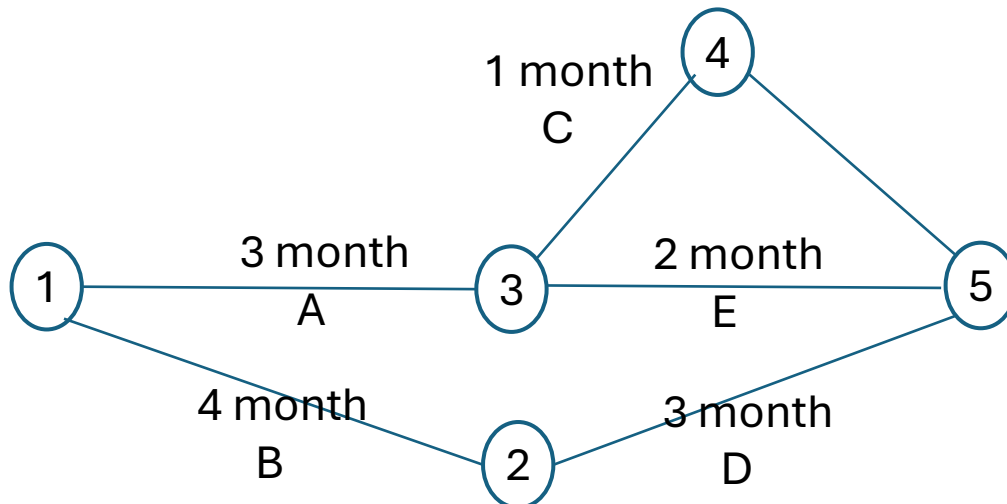


A now Diagraming Method

- در این دیاگرامی رسم میشود که یال ها بیانگر فعالیت ها میباشند.



- بر روی یال ها هزینه انجام task ها نشان داده میشود و نودها فقط ارتباط ها را نشان می دهد.



تکنیک های کاهش زمان پروژه

۱. اضافه کردن منابع جدید : taskهایی که روی مسیر بحرانی هستند را در نظر میگیریم و منابع مورد نیاز آن را زیاد میکنیم.
۲. کاهش **scope** پروژه : در صورتی که زمان کم بود انجام کارها را محدودتر میکنیم.
۳. تغییر توالی وظایف : شاید با جابه جا کردن Task ها کار سریعتر انجام شود.
۴. **Time boxing** : زمان انجام پروژه را به بازه هایی میشکنیم و افق زمانهای جزئی را مشخص میکنیم. در صورتی که انجام یک Task از این زمان بیشتر شد، آن Task متوقف شده و بقیه کارها را ادامه میدهیم.