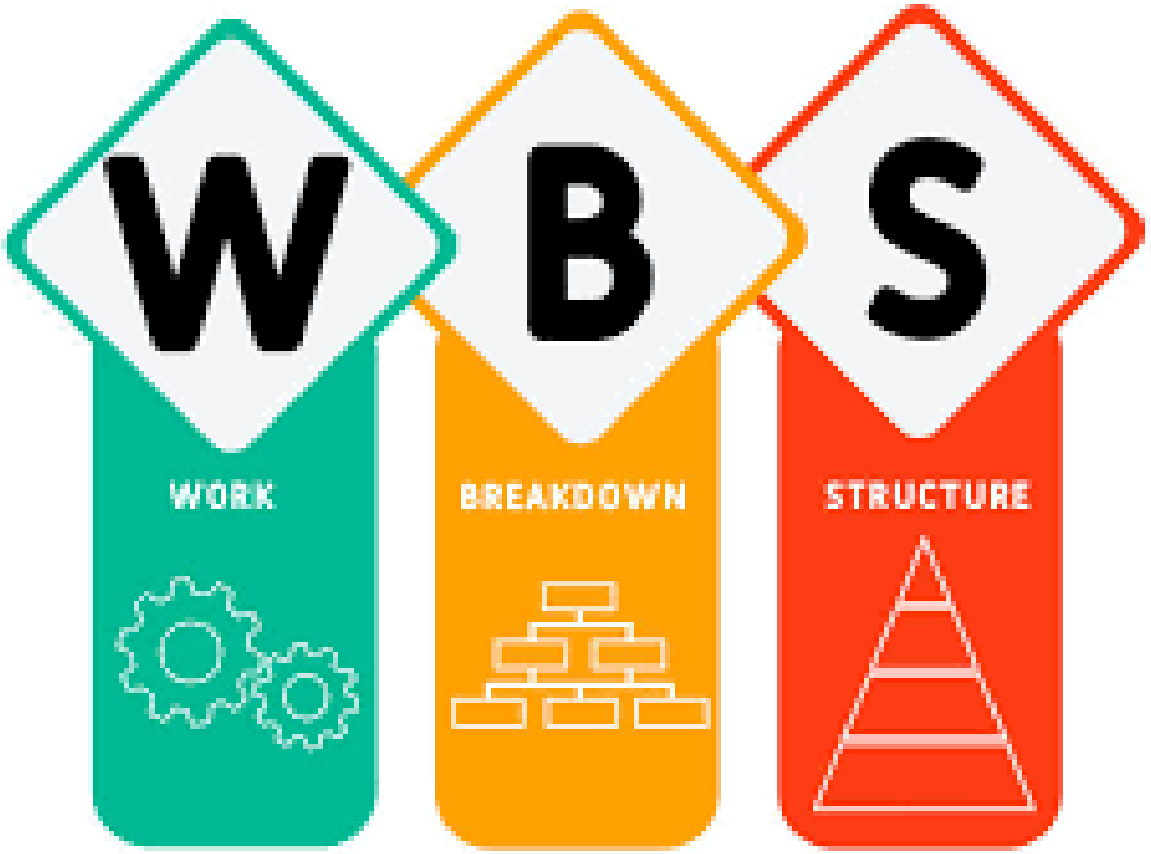
فصل ششم

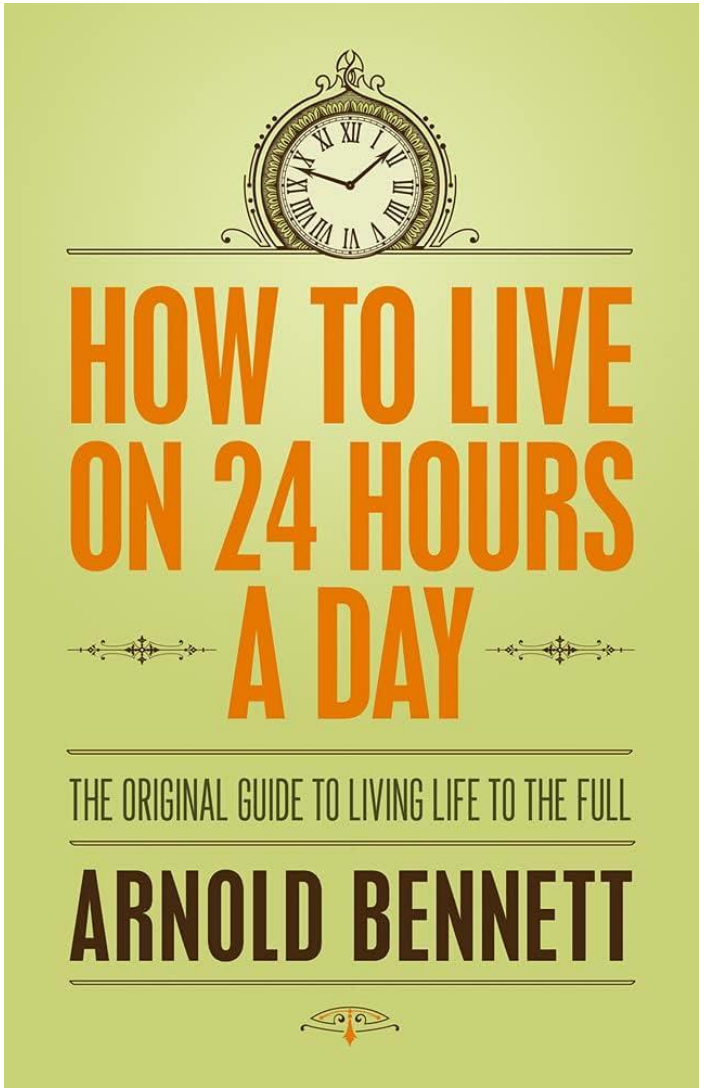
ساختار شکست کار و تخمین پروژه

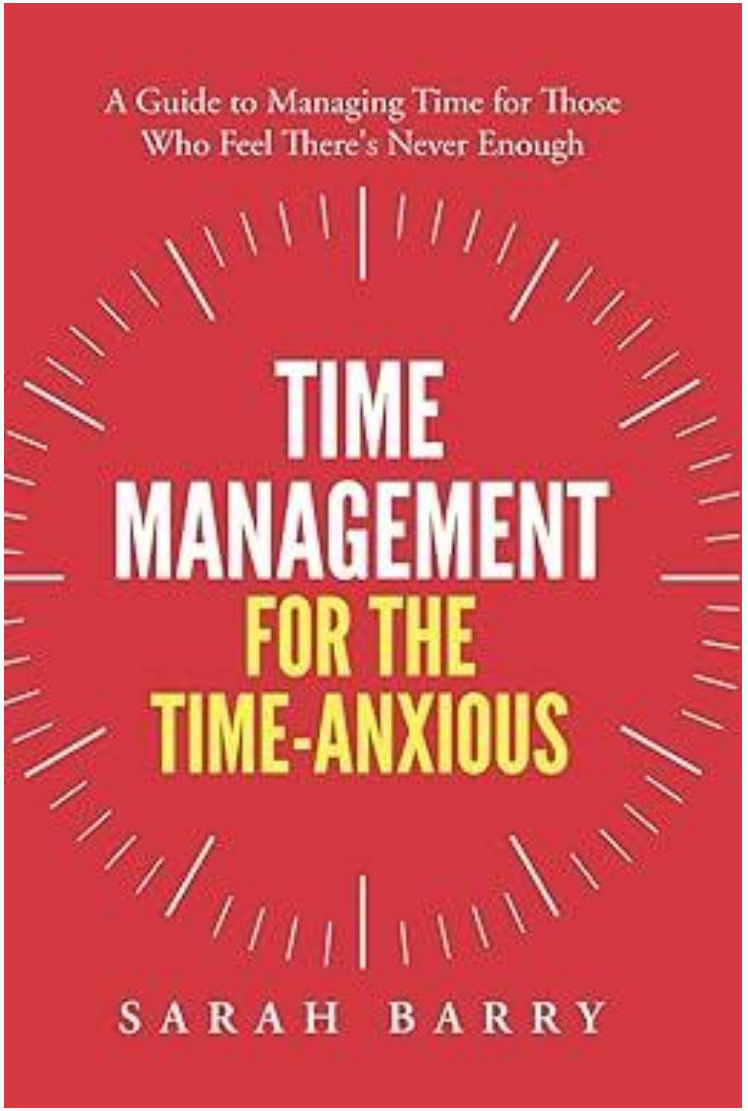


مرور فصل

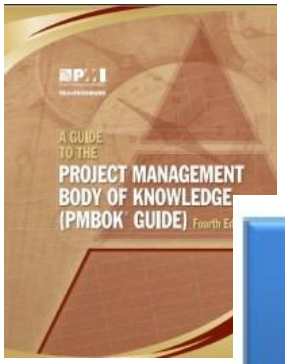
- توسعه ساختار شکست کار
- تبیین تفاوت بین اقلام تحویل دادنی و فرسنگنما
- تبیین و به کارگیری چندین روش تخمین پروژه شامل تکنیک دلفی، چارچوب بندی زمان، تخمین بالا به پایین و تخمین پایین به بالا
- تببین و به کارگیری چندین روش مهندسی نرم افزار شامل خطوط کد، تحلیل نقطه ای تابع، کوکومو و اکتشافی.

مهم ترین
دارایی ما؟


HOW TO LIVE ON 24 HOURS A DAY
THE ORIGINAL GUIDE TO LIVING LIFE TO THE FULL
ARNOLD BENNETT


A Guide to Managing Time for Those Who Feel There's Never Enough
TIME MANAGEMENT FOR THE TIME-ANXIOUS
SARAH BARRY

فاکتور های بنیادی موفقیت پروژه



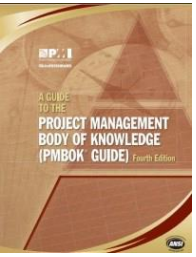
PMBOK

فرآیندها	هزینه	ریسک
زمان	یکپارچگی	ارتباطات
محدوده	منابع انسانی	کیفیت

مدیریت زمان پروژه

متمرکز بر فرآیندهای لازم برای توسعه زمانبندی پروژه و اطمینان از اتمام به موقع پروژه است.

فرآیندهای مدیریت زمان پروژه



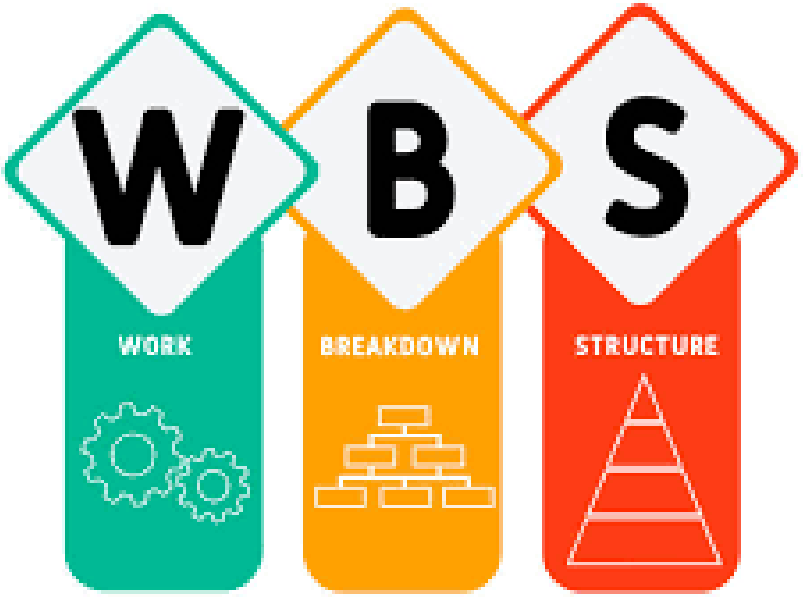
تعیین فعالیت ها	تعیین فعالیت های لازم برای تولید خروجی ها و تحویل دادنی های محدوده
تعیین توالی فعالیت ها	تعیین لزوم توالی یا موازی بودن فعالیت ها و ارتباط محتمل بین فعالیت ها
تخمین منابع فعالیت ها	تعیین منابع مورد نیاز و میزان آنها برای انجام فعالیت های پروژه
تخمین مدت زمان فعالیت ها	زمان مورد نیاز برای انجام هر فعالیت
توسعه زمانبندی	بر اساس منابع، فعالیت ها، توالی آنها و تخمین زمانی آنها یک زمان بندی برای کل بودجه توسعه داده می شود
کنترل زمانبندی	اطمینان از انجام فرآیند های صحیح برای کنترل تغییرات زمانبندی پروژه

در این فصل

- تعیین فعالیت ها و تخمین زمان آنها
- ساختار شکست کار: ساختار سلسله مراتبی برای فعالیت های پروژه
- ساختار شکست کار پلی بین محدوده پروژه و طرح جزیی پروژه است.
- استفاده از بسته های نرم افزاری زیبا و راحت اما لزوم توجه درتعیین فعالیت ها و زمان های آنها
- تخمین علم دقیقی نیست!
- تخمین به پیچیدگی **فعالیت، منابع؛ محیط** وابسته است. تخمین در **ابتدا** غیر دقیق تر است.

ساختار شکست کار

Work Breakdown Structure(WBS)



بسته های کاری

- ساختار شکست پروژه را به واحدهای کاری کوچکتر و قابل مدیریت تری به نام بسته های کار تقسیم می کند.
- هر بسته یک خروجی ملموس و قابل ارزیابی دارد
- بسته های کاری توسعه طرح پروژه، زمانبندی و کنترل پروژه را ممکن می سازند.

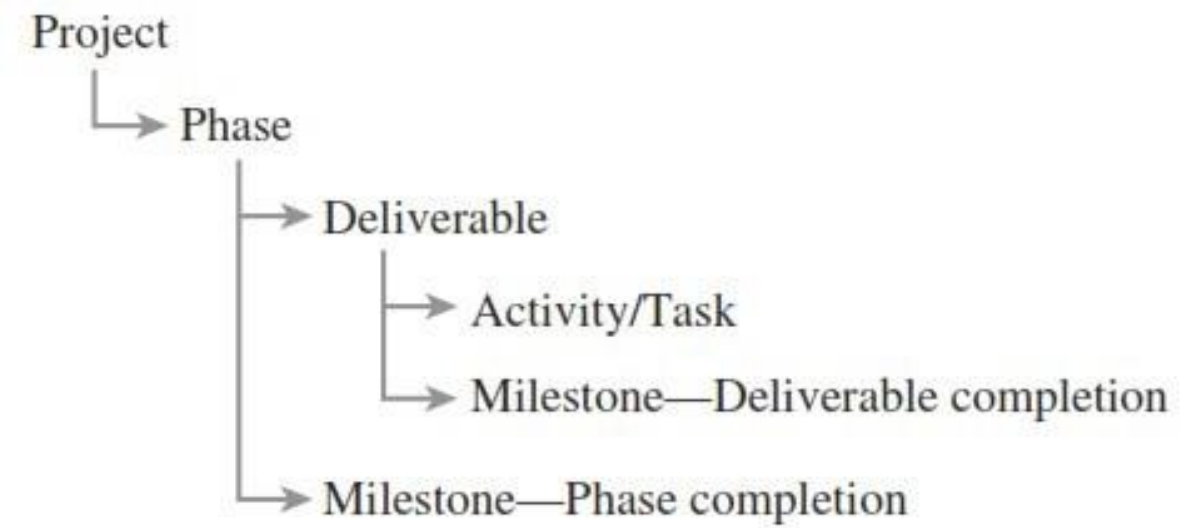
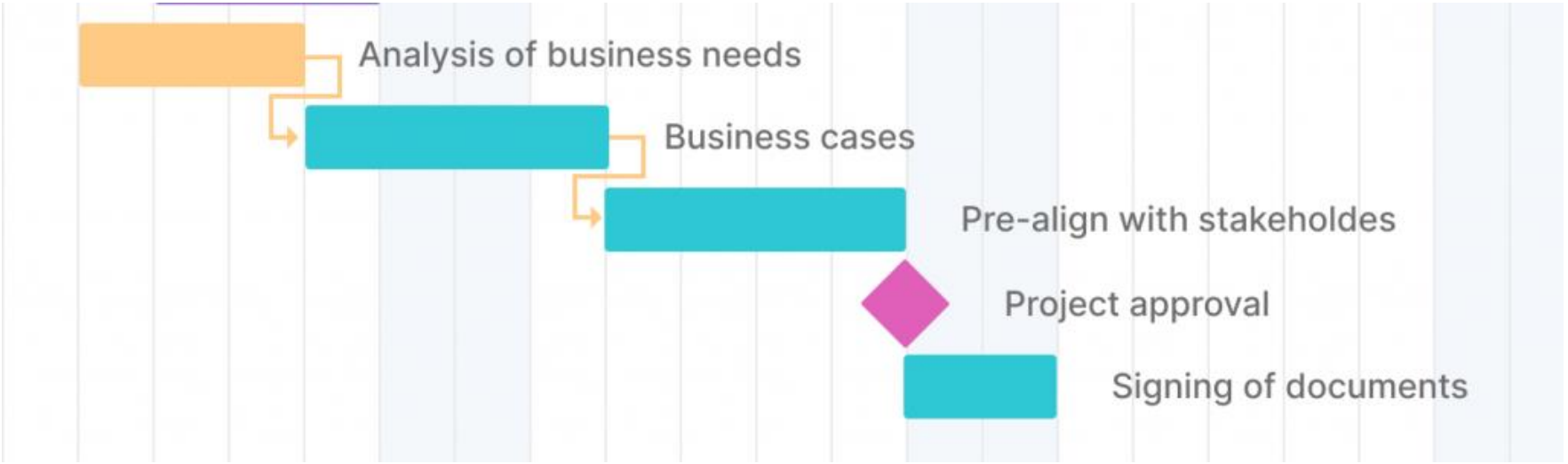


Figure 6.1 Work Package

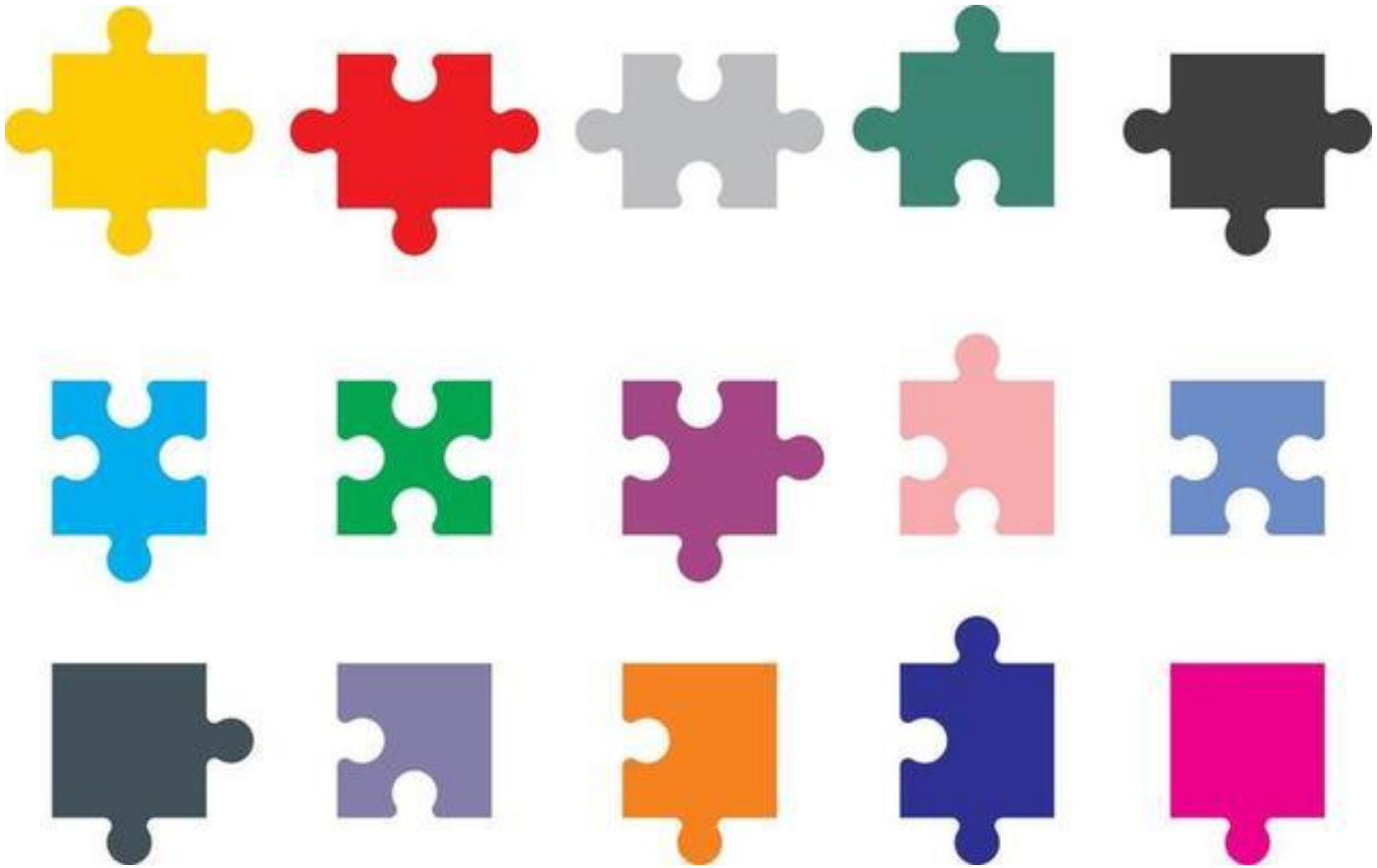
Milestone VS. Deliverable

فرسنگنما درمقابل اقلام تحویل دادنی

- فرسنگنما و تحویل دادنی ها به هم مرتبط هستند اما یکی نیستند
- تحویل دادنی ها شامل ارایه، گزارش، طرح ها و نرم افزار هستند.
- فرسنگنما بر یک دستاورد متمرکز هستند.
- فرسنگنما منجر به افزایش تمرکز تیم،کاهش ریسک، افزایش انگیزه، اطمینان از صحت ایده های نو و مبهم وکنترل کیفیت پروژه میشود.


Analysis of business needs
Business cases
Pre-align with stakeholdes
Project approval
Signing of documents

پروژه خود را چگونه به قطعات کوچک تر معنادار بشکنیم؟



توسعه ساختار شکست کار

- ممکن است چندین نسخه تهیه شود تا همه افراد از صحت و کامل بودن آن راضی گردند.
- ساختار شکست کار تا اندازه ای مبهم است و ابهام آن وابسته به ماهیت و اندازه پروژه است.

مثال موردی: سایت تجارت الکترونیک بانک

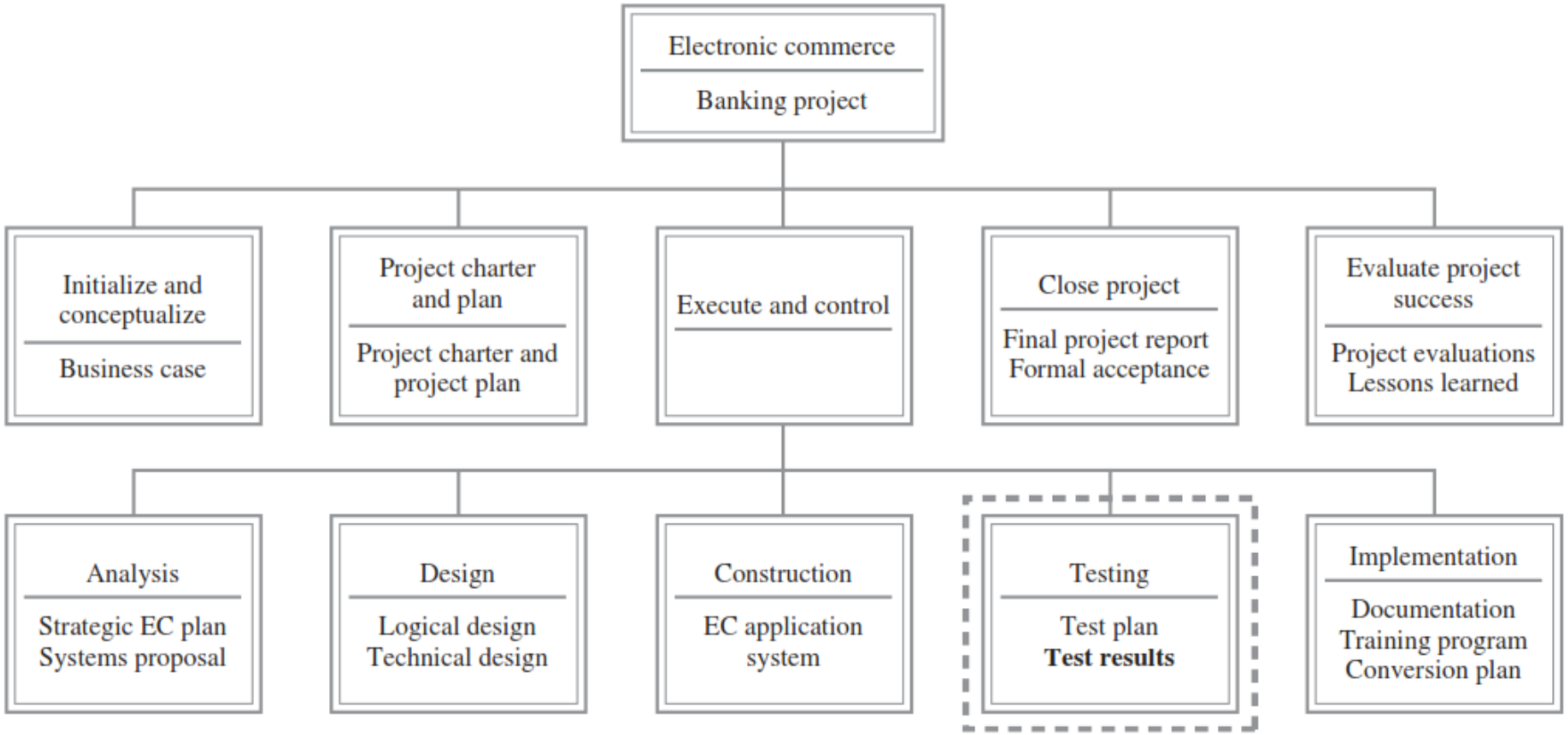


Figure 6.2 Deliverable Structure Chart (DSC) for EC Example

فعالیت های لازم برای تولید مستندات مرتبط با تست

- مرور طرح تست با مشتری به نحوی که نحوه تست، موارد مورد تست وزمان تست برای آنها مشخص گردد.
- انجام تست های بیان شده در طرح تست
- جمع آوری نتایج و تحلیل آنها
- خلاصه سازی نتایج به صورت گزارش
- در صورت تایید نتایج توسط مشتری و امضای نتایج تست، می توانیم وارد فاز بعدی، پیاده سازی، شویم در غیر اینصورت باید مشکلات را بررسی ورفع نماییم. فاز تست تنها با تهیه طرح تست و تدوین گزارش آن به انجام نمیرسد. لازم است مشتری آنها را تایید کند و صحت تطابق آن با استانداردهای کیفی ازپیش تعیین شده بررسی شود.

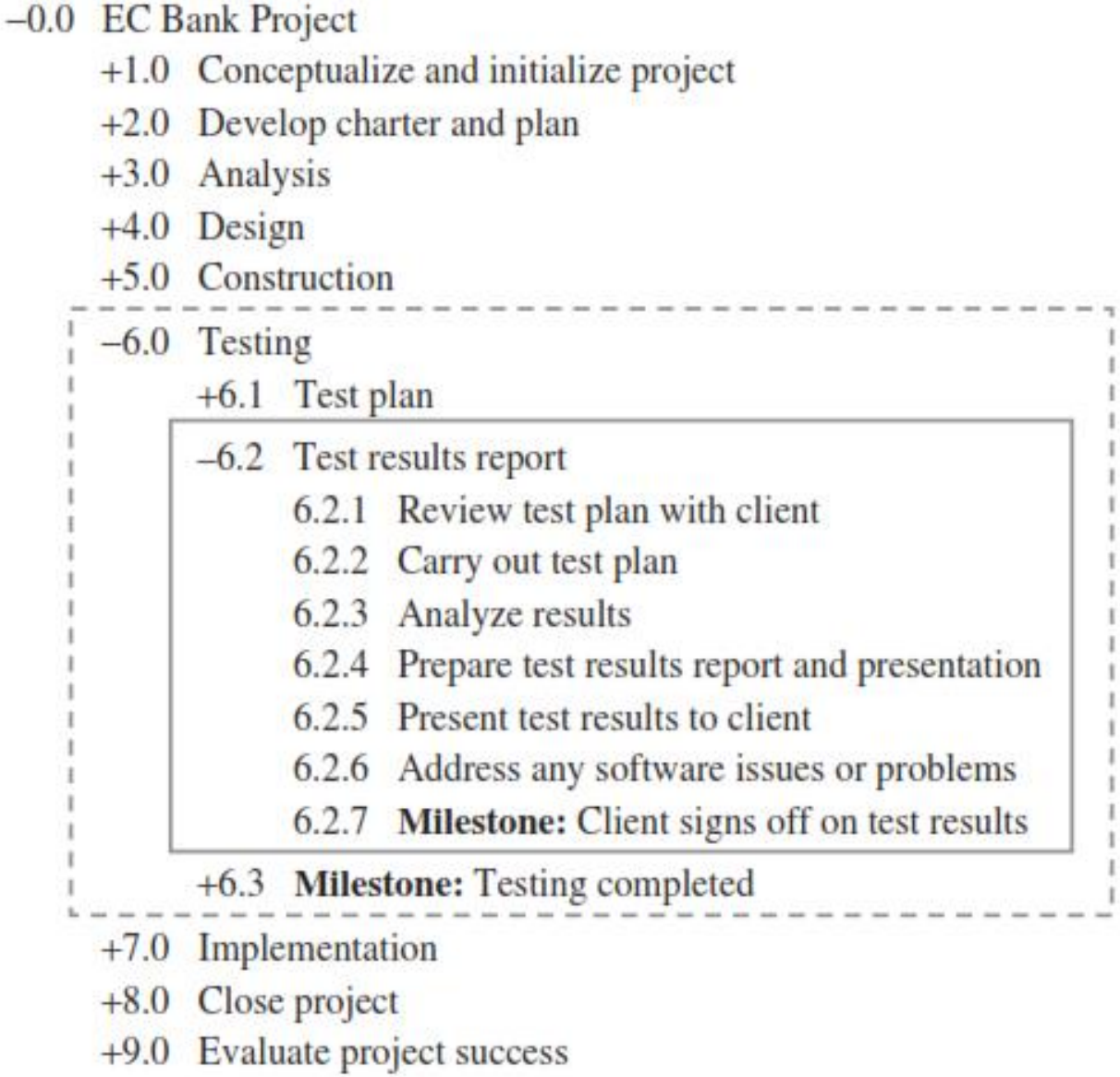
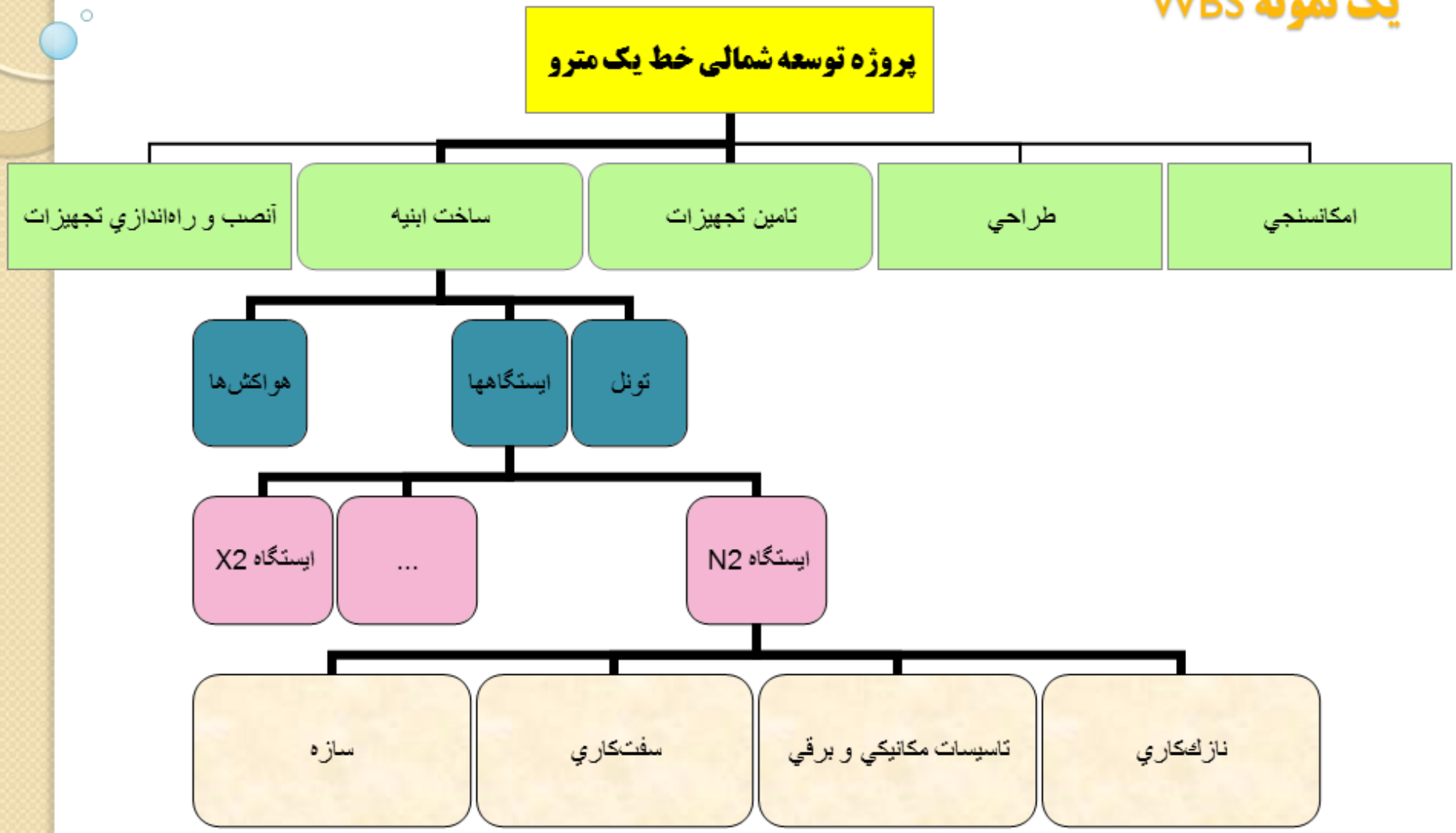


Figure 6.3 Work Breakdown Structure

یک نمونه WBS


پروژه توسعه شمالی خط یک مترو
امکانسنجي
طراحي
تامين تجهيزات
ساخت ابنيه
آنصب و راه‌اندازي تجهيزات
تونل
ايستگاهها
هواکش‌ها
ايستگاه N2
...
ايستگاه X2
نازک‌کاري
تاسيسات مکانيکي و برقي
سفت‌کاري
سازه

تعیین عناصر عمده پروژه

- تجزیه پروژه به چند عنصر یا گروه (تعیین سطح اول **WBS**)
- می‌تواند براساس **مراحل چرخه حیات پروژه** باشد. Phase Orientation Approach
- می‌تواند برمبنای **چارت سازمانی پروژه** باشد. Organization Orientation Approach
- می‌تواند برمبنای **جغرافیا و مکان اجرای پروژه** باشد. Geographical Approach
- می‌تواند برمبنای **محصول و اجزای آن** باشد. Product Orientation Approach
- می‌تواند برمبنای **زیر پروژه‌ها** باشد. Project Orientation Approach

نکات توسعه ساختار شکست کار

- **ساختار شکست کار باید تولید محور باشد نه تنها متمرکز بر انجام تعدادی فعالیت**
- **ساختار شکست کار باید هم راستا با آسا(MOV)پروژه باشد.**

– هماهمنگ با اقلام تحویل دادنی محدوده

– قانون ۱۰۰ درصد (نتیجه این قانون اطمینان از لحاظ کردن تمام فعالیت ها و منابع هزینه ای است)

- **سطح بیان جزییات باید اجازه کنترل و طرح ریزی پروژه را بدهد.**

–متداول ترین خطاهای توسعه ساختار شکست کار: جزییات بسیار زیاد یا بسیار کم

جزییات کم: حذف فعالیتهای اصلی و زمانبندی و بودجه خوش بینانه

جزییات زیاد: لیست کار ساعتی- بیش مدیریتی و در نتیجه کاهش روحیه تیم و دشواری ردیابی فعالیت ها

- **افرادی که قرار است فعالیت ها را انجام دهند بایستی در فرآیند توسعه ساختار شکست کار مشارکت داشته باشند.**

- افراد متخصص هر حوزه درک بهتری از سطح جزییات مناسب آن حوزه را دارند.
- افزایش حس مشارکت و همکاری در صورت مشارکت در توسعه ساختار
- توضیح بسته های کاری در لغت نامه WBSبرای جلوگیری از اغتشاش و کج فهمی

Haugan (2002) also suggests that the **100 percent rule** is the most important criterion in developing and evaluating the WBS. The rule states:

"The next level decomposition of a WBS element (child level) must represent 100 percent of the work applicable to the next higher(parent) element."

ارزیابی پروژه

- پس از تعیین فعالیت ها کام بعد تخمین مدت زمان انجام فعالیت است.

یکی از سخت ترین و اساسی ترین فعالیت های مدیریت پروژه ارزیابی زمان انجام یک فعالیت به خصوص است.

- ارتباط مستقیمی با بودجه دارد
- روشهای ارزیابی فعالیت ها :

– حدسی

– دلفی

– بالا به پایین

– پایین به بالا

As T. Capers Jones (Jones 1998) points out
The seeds of major software disasters are usually sown in the first three months of commencing the software project. Hasty scheduling, irrational commitments, unprofessional estimating techniques, and carelessness of the project management function are the factors that tend to introduce terminal problems. Once a project blindly lurches forward toward an impossible delivery date, the rest of the disaster will occur almost inevitably .

GUESSTIMATING

روش حدسی

- لطفا عددی را از داخل کلاه بیرون بیاورید!
- تجربه کم گاهی مدیر پروژه را وادار می کند زمان انجام پروژه را حدس بزند و بر زمان را بر اساس احساسات تعیین کند و نه بر اساس مستندات محکم
- روشی سریع و آسان
- افراد غالبا بیش از اندازه خوش بین هستند->

– تلاش کنید تا بدست آوردن اطلاعات کافی صبر کنید.

– بازه اطمینان برای هر فعالیت تعیین کنید.

مثال: سه تا شش ماه و

۳۰ تا ۶۰ هزار دلار

روش دلفی

- روش دلفی از چندین خبره برای رسیدن به اجماع در مورد موضوع خاصی استفاده می کند.
- برای این کارباید از چندین خبره برای ارزیابی زمان مشخصی سوال می شود. نظرات آنها مقایسه میشود اگر نزدیک به هم بود میانگین گرفته می شود و اگر تفاوت زیادی داشت، بحث و تبادل نظر می شود تا خبرگان پس از آن ارزیابی جدیدی از زمان ارایه دهند. این کار تا رسیدن به اجماع ممکن است چندین بار تکرار شود.
- روش دلفی تقریبا از تمامی دیگر روش ها زمان بیشتری به طول می انجامد اما می تواتد بسیار موثر ، مطمئن و با حاشیه کمی از خطا همراه باشد.
- چه زمانی از این روش استفاده می شود: وقتی خبرگان زیاد باشند و رسیدن به اجماع دشوار و یا وقتی خبرگان به جهت جغرافیایی از هم دور باشند.

Time Boxing

روش چارچوب بندی زمان

- چارچوب بندی زمان یعنی اختصاص **بازه زمانی** مشخص برای انجام دادن کارهایی از پیش تعیین شده، در واقع بجای اینکه تا پایان یافتن کاری وقت بگذارید بایستی متعهد شوید که تمام یا بخشی از کار را در مدت زمانی معلوم انجام دهید و نباید این برایتان مهم باشد که کل کار پایان پذیرد.

– راهکارها: جلوگیری از تعلل، پرهیز از کمال گرایی، شکست کارهای بزرگ به کارهای کوچکتر، ایجاد ضرب آهنگ کار و ....

- منجر به افزایش تمرکز بر کارهای مهم می شود.
- افزایش بی رویه این روش منجر به خستگی

و دل زدگی افراد تیم می شود

تخمین بالا به پایین

- تخمین زمان و یا هزینه بر مبنای مدت زمانی که آن فعالیت **باید** به طول انجامد و هزینه ای که **باید** داشته باشد.
- این محدودیت می تواند حاصل از یک دستور بالادستی، طرح استراتژیک بالا دستی، شرایط بازار و رقبا باشد(مثال).
- پس از تعیین هدف کلی مدیریت پروژه باید درصدی از زمان و منابع را به هر فعالیت تخصیص دهد.
- اگر هدف کلی منطقی، واقع گرایانه و دست یافتنی باشد این روش موفق خواهد بود.
- رژه مرگ

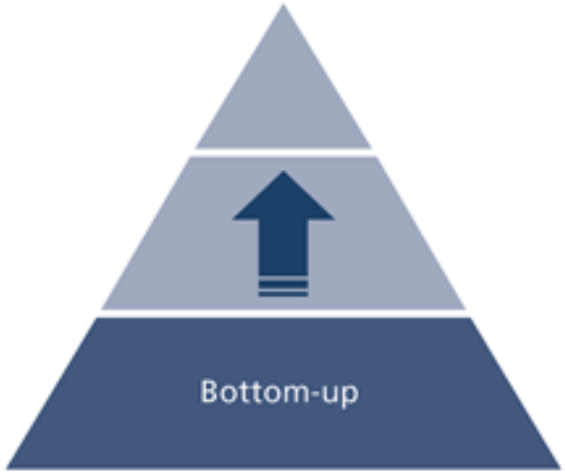
رژه مرگ

٪۵۰

- رژه مرگ!(یوردان، ۱۹۹۹)

– اگر پارامترهای پروژه ۵۰ درصد از نرمال پیشی بگیرند.

پارامترهای پروژه شامل زمانبندی، بودجه، نیروی انسانی، عملکرد، ویژگی ها و یا دیگر نیازمندی های فنی و عملکردی است.



روش پایین به بالا

- غالب تخمین های در دنیای واقعی با این روش صورت میگیرد.
- شکستن پروژه به فعالیت های کوچکتر و تخمین زمان و منابع آن فعالیت ها بر مبنای نفر-ساعت، نفر- هفته و یا نفر-ماه.
- ساختار شکست کار زیربنایی برای این روش ایجاد می کند.
- با لیستی از فعالیت های مورد انتظار شروع می شود. سپس زمان لازم برای هر فعالیت ارزیابی می شود.

روش پایین به بالا

- ارزیابی فعالیت ها تابعی از خود فعالیت، منابع و پشتیبانی ها هستند.
  - فعالیت: ماهیت و پیچیدگی و...
  - منابع: تجربه، انگیزه و سطح خبرگی نیروی انسانی
  - پشتیبانی: فناوری، ابزار، محیط کاری و ...

6.2	Test results report	
	6.2.1. Review test plan with client	1 day
	6.2.2. Carry out test plan	5 days
	6.2.3. Analyze results	2 days
	6.2.4. Prepare test results report and presentation	3 days
	6.2.5. Present test results to client	1 day
	6.2.6. Address any software issues or problems	5 days

سنجه ها و روشهای مهندسی نرم افزار

مهم ترین چالش پروژه فا تخمین زمان و نیروی لازم برای تحویل مهم ترین قلم تحویل دادنی یعنی سیستم نرم افزاری است.

- چالش، ارزیابی کاری است که بیشتر ذهنی است تا فیزیکی و تا مراحل نهایی چرخه حیات پروژه به خوبی تعریف نمی شود.
- تعریف محدوده یک نگاه سطح بالا به نرم افزار ارایه می دهد و مرز پروژه را تعیین می نماید.
- نیازمندی های دقیق مساله و ویژگی ها و عملکردهای خاص پروژه تنها در حین فرآیند طراحی تعیین می گردند. علاوه بر این پیچیدگی ها و چالش های فنی درابتدا نامعین هستند و معمولا سرهم بندانه اعلام می شوند!

تخمین پروژه فا مانند تلاش برای زدن به هدفی در حال حرکت است و نیازمند تنظیم پیوسته است.

(Lines Of Code)
تعداد خطوط کد

- متداول ترین وسنتی ترین سنجه برای ارزیابی اندازه نرم افزار
- چه چیز هایی را به عنوان خطوط کد در نظرمی گیریم؟
  - خطوط کامنت را در نظر بگیریم؟
  - تعریف متغیرها را در نظر بگیریم؟
  - تفاوت کدنویسان حرفه ای و مبتدی
  - تفاوت تعداد خطوط در زبانهای برنامه نویسی
  - کاهش بهره وری
  - شمارش بعد از نوشتن کد و یا قبل از آن؟

نقاط عملکردی
Function Point (FP)

- معیاری برای ارزیابی عملکرد و پیچیدگی یک سیستم نرم افزاری است.

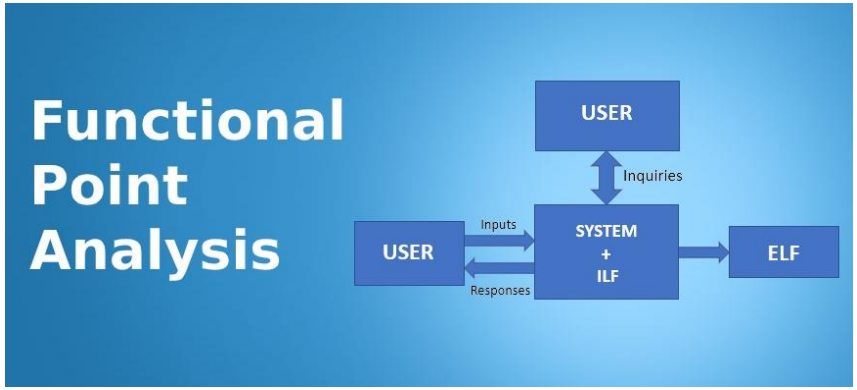
500 FP< 1000 FP

- مزیت این روش :

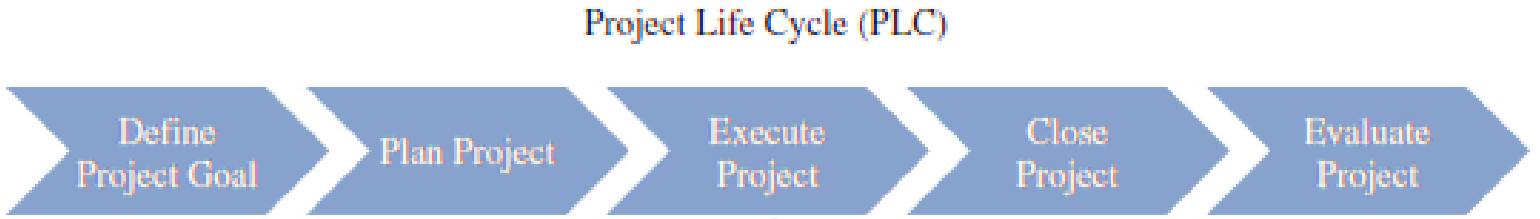
– مستقل از فناوری و زبان

– قابل اعتماد: توسط دو فرد متفاوت نتایجی مشابه با خطای قابل قبول به دست می آید.

- کلید شمارش تعداد نقاط عملکردی، **فهم نیاز کاربر** است.



- تعداد نقاط عملکردی پروژه را در مراحل مختلف چرخه پروژه بایستی محاسبه کرد. ابتدا پس از تعیین محدوده پروژه محاسبه می شود اما این مقدار مجددا در فازهای تحلیل و طراحی نیز محاسبه می گردد.



نقاط عملکردی
Function Point

تحلیل نقاط عملکردی بر مبنای ارزیابی پنج نوع داده و تراکنش است که مرز نرم افزار را مشخص می کنند.

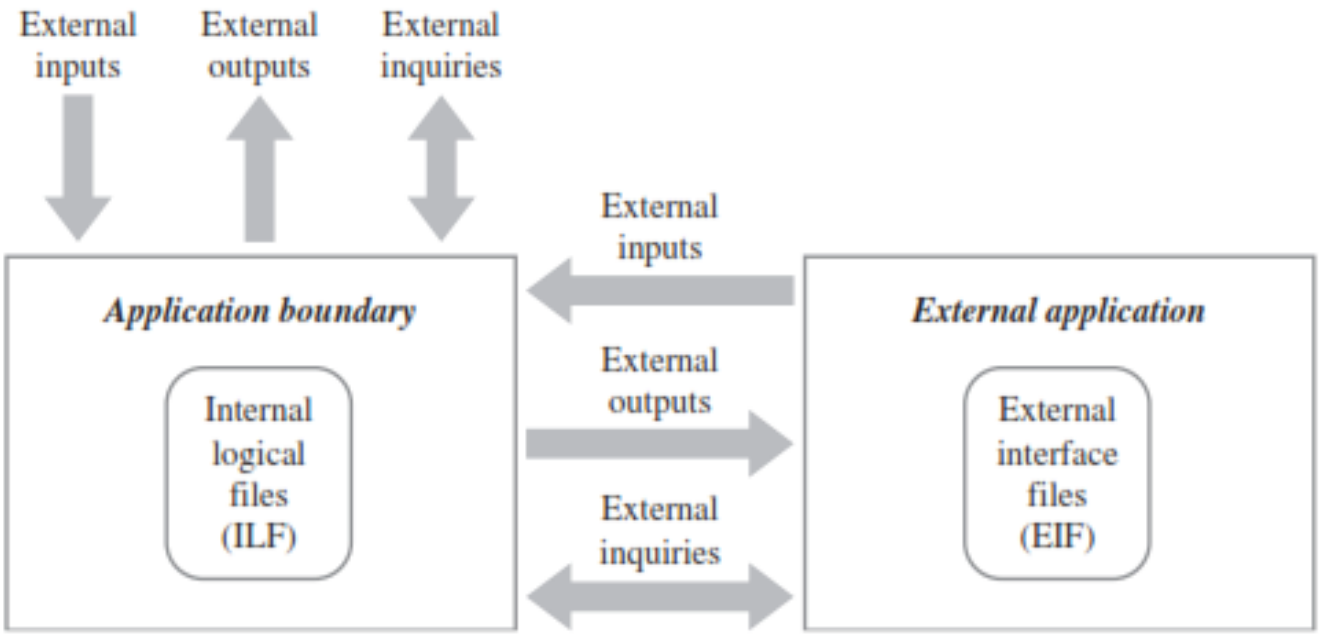
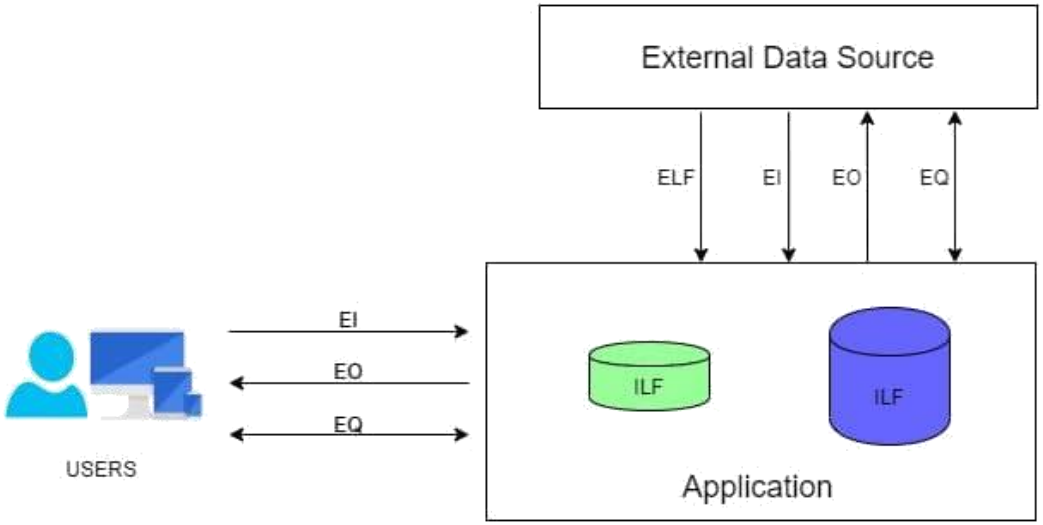


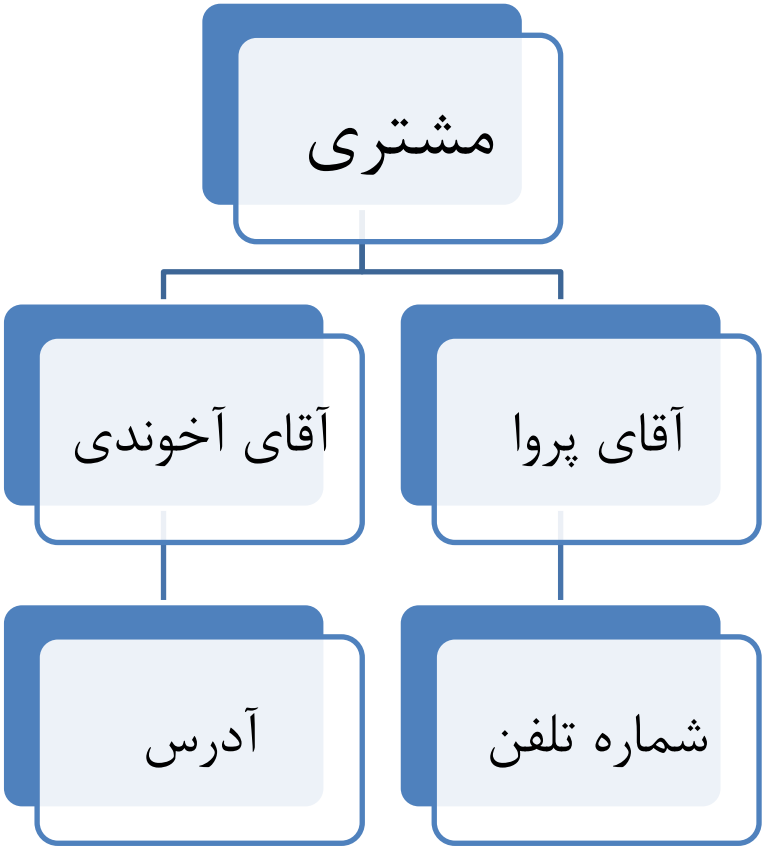
Figure 6.5 The Application Boundary for Function Point Analysis

فایلهای منطقی درونی
ILF

- یک فایل منطقی که داده ها را **درون مرز نرم افزار** ذخیره می کند.
- مثال: هر یک از موجودیت های ERD
- پیچیدگی فایلها بسته به عناصر داده ای و زیرگروههای داده ای که در فایلها نگهداری می شوند به صورت کم، متوسط یا زیاد بیان می شوند.



نمونه داده ، گروه داده ای و زیر گروه داده



گروه داده ای

زیرگروه داده ای

داده

نقاط عملکردی
Function Point

تحلیل نقاط عملکردی بر مبنای ارزیابی پنج نوع داده و تراکنش است که مرز نرم افزار را مشخص می کنند.

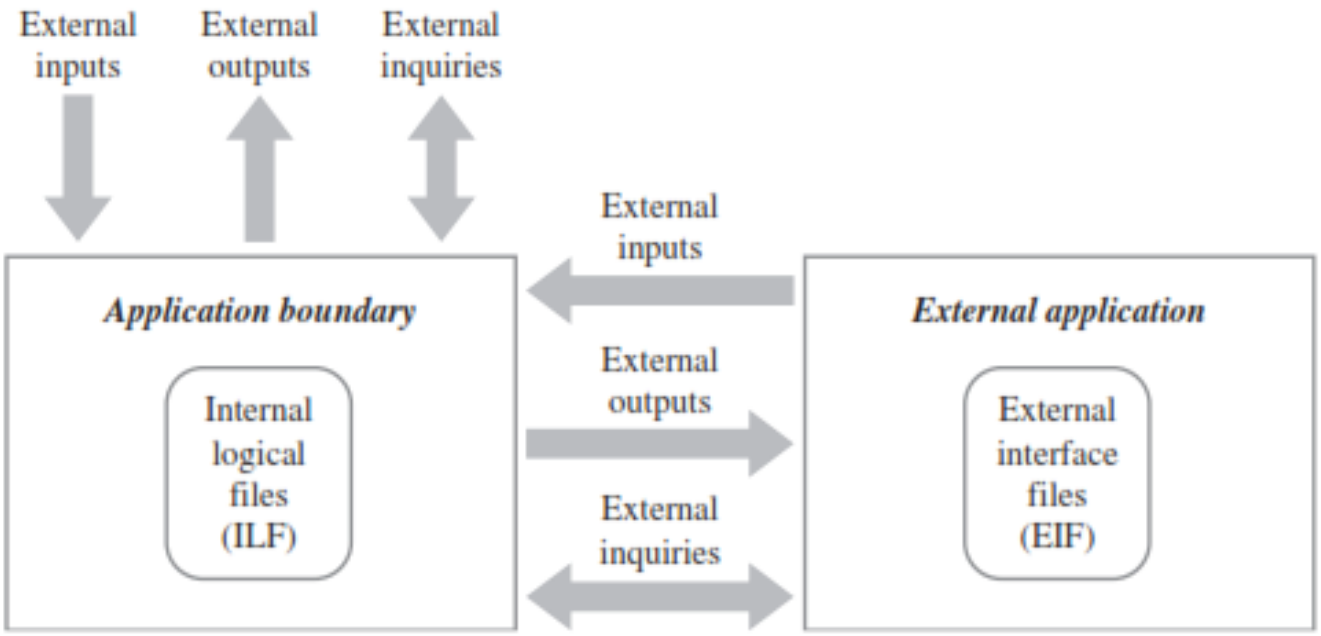


Figure 6.5 The Application Boundary for Function Point Analysis

فایلهای واسط بیرونی
EIF

- مانند ILF با این تفاوت که این فایلها توسط یک سیستم بیرونی نگهداری می شود.

ورودی بیرونی
EI

- ورودی بیرونی: **به فرآیند و یا تراکنش های داده ای** گفته می شود که مبدا آن خارج مرز سیستم است و با عبور از مرز سیستم وارد سیستم می شود. نتیجه ان معمولا به صورت اضافه، حذف، بروزرسانی داده در یک یا تعدادی از فایلهای داخلی است. مثال: ورود داده از طریق واسط کاربری به سیستم با کیبرد، بین سیستم ها، مثال :پرداخت الکترونیک.
- دسته بندی بر اساس پیچیدگی (تعداد فایلهای داخلی ارجاعی، تعداد عناصر داده ای و ...)

خروجی بیرونی

- مانند ورودی های بیرونی فرآیند یا تراکنشی است که اجازه خروج داده از مرز سیستم را می دهد . مثال: گزارش ها، پیامدهای تایید، گرافها، نمودارها.
- این داده ها در صفحه نمایش داده می شوند یا پرینت می شوند و یا به سیستم های دیگری ارسال می شوند.
- پس از شمارش تعداد خروجی های بیرونی بر اساس میزان پیچیدگی(به روشی مانند ورودی بیرونی) رتبه بندی می شوند.

پرس و جوهای بیرونی

فرآیند یا تراکنش هایی که شامل ترکیبی از ورودی ها و خروجی ها برای بازیابی داده از فایلهای داخلی و یا فایلهای خارجی به نرم افزار است. **پرس و جوها در داده های ذخیره شده تغییری ایجاد نمیکنند و تنها این داده ها را می خوانند**. پرس و جوهایی با منطق پردازشی متفاوت و یا فرمت ورودی و خروجی متفاوت همه به صورت یک پرس و جو شمرده می شوند. پس از شناسایی این پرس و جوها، بر اساس تعداد عناصر داده ای و یا تعداد فایلهای ارجاع داده شده به سه رده کم، متوسط، زیاد از لحاظ پیچیدگی رتبه بندی می شوند.

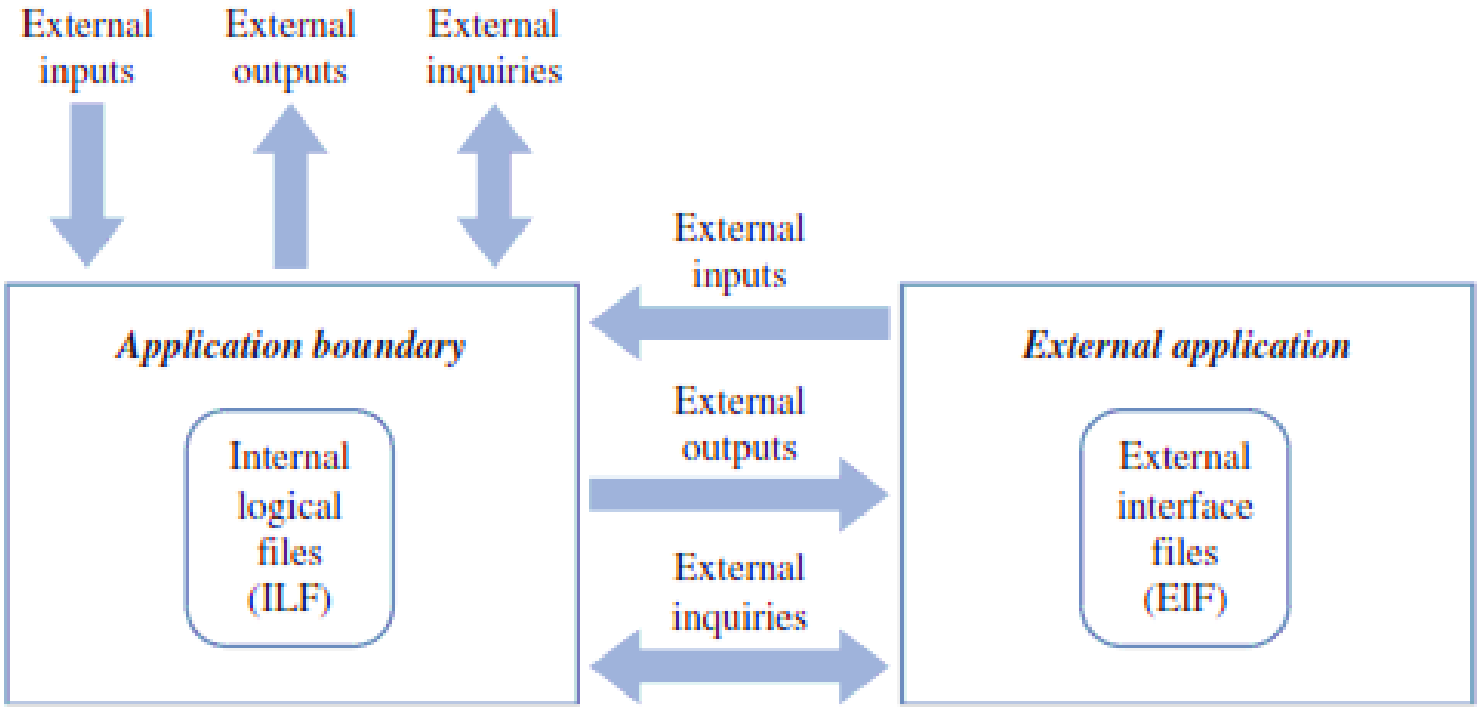


Figure 6.5 The Application Boundary for Function Point Analysis

Type of Component	Complexity of Components			
	Low	Average	High	Total
External Inputs	___ x 3 = ___	___ x 4 = ___	___ x 6 = ___	
External Outputs	___ x 4 = ___	___ x 5 = ___	___ x 7 = ___	
External Inquiries	___ x 3 = ___	___ x 4 = ___	___ x 6 = ___	
Internal Logical Files	___ x 7 = ___	___ x 10 = ___	___ x 15 = ___	
External Interface Files	___ x 5 = ___	___ x 7 = ___	___ x 10 = ___	

Unadjusted Function Point(UFP)

- *ILF:* 3 Low, 2 Average, 1 Complex
- *EIF:* 2 Average
- *EI:* 3 Low, 5 Average, 4 Complex
- *EO:* 4 Low, 2 Average, 1 Complex
- *EQ:* 2 Low, 5 Average, 3 Complex

Table 6.1 Computing UAF

	Complexity			
	Low	***Average***	***High***	***Total***
Internal logical files (ILF)	3 × 7 = 21	2 × 10 = 20	1 × 15 = 15	56
External interface files (EIF)	___ × 5 = ___	2 × 7 = 14	___ × 10 = ___	14
External input (EI)	3 × 3 = 9	5 × 4 = 20	4 × 6 = 24	53
External output (EO)	4 × 4 = 16	2 × 5 = 10	1 × 7 = 7	33
External inquiry (EQ)	2 × 3 = 6	5 × 4 = 20	3 × 6 = 18	44
Total unadjusted function points (UAF)				200

Value Adjustment Factor(VAF)

Degree of Influence(DI)—Processing Complexity Adjustment(PCA)

- عددی بین ۰ تا ۵ به هریک منتصب نمایید
- عدد بالاتر نشان دهنده اهمیت بیشتر

Table 6.2 GSC and Total Adjusted Function Point

General System Characteristic	*Degree of Influence*
Data communications	3
Distributed data processing	2
Performance	4
Heavily used configuration	3
Transaction rate	3
On-line data entry	4
End user efficiency	4
Online update	3
Complex processing	3
Reusability	2
Installation ease	3
Operational ease	3
Multiple sites	1
Facilitate change	2
Total degrees of influence (TDI)	40
VALUE ADJUSTMENT FACTOR VAF = (TDI * 0.01) +.65	VAF = (40 *.01) +.65 = 1.05
Total adjusted function points = FP = UAF * VAF	FP = 200 * 1.05 = 210

SOURCE: Adapted from Dennis and Haley (2000).

- 0 = not present or no influence
- 1 = incidental influence
- 2 = moderate influence
- 3 = average influence
- 4 = significant influence
- 5 = strong influence

General System Characteristic		Brief Description
1.	Data communications	How many communication facilities are there to aid in the transfer or exchange of information with the application or system?
2.	Distributed data processing	How are distributed data and processing functions handled?
3.	Performance	Was response time or throughput required by the user?
4.	Heavily used configuration	How heavily used is the current hardware platform where the application will be executed?
5.	Transaction rate	How frequently are transactions executed daily, weekly, monthly, etc.?
6.	On-Line data entry	What percentage of the information is entered On-Line?
7.	End-user efficiency	Was the application designed for end-user efficiency?
8.	On-Line update	How many ILF's are updated by On-Line transaction?
9.	Complex processing	Does the application have extensive logical or mathematical processing?
10.	Reusability	Was the application developed to meet one or many user's needs?
11.	Installation ease	How difficult is conversion and installation?
12.	Operational ease	How effective and/or automated are start-up, back-up, and recovery procedures?
13.	Multiple sites	Was the application specifically designed, developed, and supported to be installed at multiple sites for multiple organizations?
14.	Facilitate change	Was the application specifically designed, developed, and supported to facilitate change?

با عدد Adjusted FP چه کنیم؟

- روش اول: بهره وری، هر فرد توانایی تولید چه تعداد FP را در یک زمان مشخص، مانند روز، سال، ماه، دارد؟

  استفاده از داده های مشابه پروژه های قبلی برای تخمین بهتر

- روش دوم: تبدیل مقدار FPبه LOC(تعداد خطوط کد).

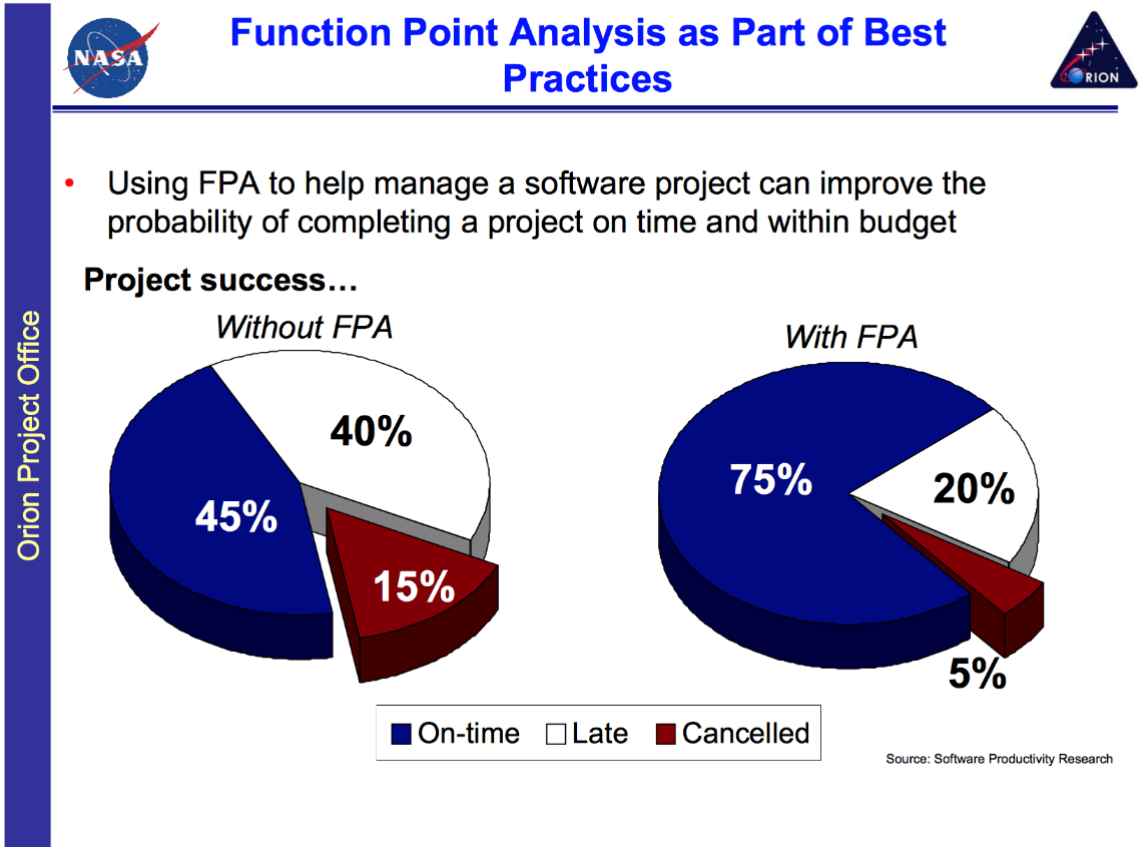
Table 6.3 Function Point Conversion to LOC

Language	*Average Source LOC per Function Point*	*Average Source LOC for a 210 FP Application*
Basic	107	22,470
Visual Basic 5	29	6,090
C	128	26,880
C++	53	11,130
COBOL	107	22,470
Java	53	11,130
Machine Language	640	134,440
1st Generation (default)	320	67,200
2nd Generation (default)	107	22,470
3rd Generation (default)	80	16,800
4th Generation (default)	20	4,200
5th Generation (default)	5	1,050

SOURCE: Original: http://www.theadvisors.com/langcomparison.htm. A more recent programming language table can be found at http://www.qsm.com/FPGearing.html

https://www.qsm.com/resources/function-point-languages-table

نمونه موردی استفاده از نقاط عملکردی



http://galorath.com/wp-content/uploads/2014/08/SW_5_-_Robert_G_and_Terry_V_-_Function_Point_Analysis_for_the_NASA_CEVGN.pdf

قانون بروکس

اضافه کردن نیروی انسانی به پروژه دیرهنگام، منجر به تاخیر بیشتر میشود

QUICK THINKING—MORE PEOPLE = MORE PROBLEMS

The classic book, *The Mythical Man-Month*, by Fredrick P. Brooks was first published in 1975 with an anniversary edition published 20 years later (due to the fact that some things have not changed). Brooks worked at IBM as the manager of a large project that developed the OS/360 operating system. Although the OS/360 operating system was eventually a successful product for IBM, the system was late, and cost several times more than originally estimated. In fact, the product did not perform well until after several releases. Based on his experience, Brooks wrote a number of essays that were embodied in his book. One timeless insight became known as **Brooks' Law**:

> Adding manpower to a late software project makes it later.

More recently, a study conducted by Quantitative Software Management (QSM) reports having a larger team on a software project can add millions of dollars to the cost of the project while only reducing the schedule by a few days. The study suggests that larger teams tend to create more defects or "bugs," and the additional rework detracts from any potential schedule benefits that may arise from having more people.

1. Why would adding more people to an already late IT project make it even later?
2. Many projects tend to be overstaffed during the planning and requirements gathering stages of the project. Why would having a smaller or leaner project team be a better approach?

SOURCE: Adapted from Frederick P. Brooks, *The Mythical Man-Month: Essays on Software Engineering*, 2nd ed., 1975; More People, More Bugs, from *Projects@Work*, October 6, 2005.

COCOMO
(Constructive Cost Model)

بر مبنای تعداد خطوط کد است.

برای تخمین هزینه، زمان و نیروی لازم ابتدا نوع پروژه تعیین می شود:

– **ارگانیک:**پروژه هایی روتین که فناوری، فرآیند و انسان ها بدون مشکلی در کنار هم کار می کنند. پروژه های آسان که انتظار می رود با مشکلات محدودی مواجه شوند.

– **جاسازی شده:** یک پروژه چالشی. به عنوان مثال سیستمی برای پشتیبانی از یک فرآیند جدید کسب و کار. افراد **کم تجربه تر** هستند و **سیستم و فرآیندها به بلوغ** کامل نرسیده اند.

– **نیمه–منفصل:** بین ساختار های ارگانیک و جاسازی شده. این پروژه ها ممکن است ساده و سرراست نباشند اما سازمان از فراهم بودن فناوری و فرآیند های مناسب برای برخورد با پروژه مطمئن است.

COCOMO
(Constructive Cost Model)

- مدل اولیه کوکومو با بهره گیری از یک فرمول تعداد نفر-ماه نیـروی لازم برای انواع پروژه ها را تخمین میزند. نفر-ماه، تلاش یک ماه یـک نیـروی انسانی است. در مدل کوکومو یک نفر مـاه معـادل ۱۵۲ سـاعت در نظـر گرفته شده است. بعد از تعیین نوع پروژه، نفر-ماه پروژه بر اساس فرمـول زیر محاسبه می شود.

- Organic: Person-Months $= 2.4 \times \text{KDSI}^{1.05}$
- Semi-Detached: Person-Months $= 3.0 \times \text{KDSI}^{1.12}$
- Embedded: Person-Months $= 3.6 \times \text{KDSI}^{1.20}$

Where: KDSI = Thousands of delivered source instructions, i.e., LOC

مثال از مدل کوکومو

- فرض کنید نرم افزاری با زبان جاوا با ۲۰۰ نقطه عملکردی توسعه می دهیم. به کمک جدول ۶.۳ تعداد خطوط کد ۱۰۶۰۰ درنظر گرفته می شود. اگر پروژه از درجه سختی متوسطی برخوردار باشد، نیمه منفصل خواهد بود.

$$\begin{aligned} \text{Person}-\text{Months} &= 3.0 \times \text{KDSI}^{1.12} \\ &= 3.0 \times (10.6)^{1.12} \\ &= 42.21 \end{aligned}$$

تعیین مدت زمان در مدل کوکومو

بنابراین ۴۲.۲۱ نفر-ماه برای اتمام پروژه نیاز است. برای به دست آوردن زمان لازم برای انجام پروژه از فرمول بروکس(۱۹۹۵) استفاده می کنیم که زیاد شدن نیروی انسانی را لزوما منجر به افزایش بهره وری نمی داند چرا که ارتباطات آنها پروژه را پیچیده می نماید و سرعت آن را کند می نماید. به همین جهت مدت زمان انجام پروژه به صورت زیر محاسبه می شود:

- Organic: $\text{Duration} = 2.5 \times \text{Effort}^{0.38}$
- Semi-Detached: $\text{Duration} = 2.5 \times \text{Effort}^{0.35}$
- Embedded: $\text{Duration} = 2.5 \times \text{Effort}^{0.32}$

$$\begin{aligned}\text{Duration} &= 2.5 \times \text{Effort}^{0.35} \\ &= 2.5 \times (42.21)^{0.35} \\ &= 9.26 \text{ months}\end{aligned}$$

- تعداد نیروی لازم برابر استخدام در پروژه برابر خواهد بود با:

$$\begin{aligned}\text{People Required} &= \text{Effort} \div \text{Duration} \\ &= 42.21 \div 9.26 \\ &= 4.55\end{aligned}$$

فرآیند تخمین تعداد نیروهای لازم به کمک روش کوکومو

قواعد تجربی(اکتشافی)

- بر اساس تجربیات قبلی در مورد پروژه های بعدی قضاوت مینماییم.
- فعالیت های اولیه مشابهی در هر پروژه نرم افزاری وجود دارند و این فعالیت ها درصد قابل پیش بینی ای از کل نیروی لازم برای پروژه را نیاز دارند. به عنوان مثال:

- 30 percent planning
- 20 percent coding
- 25 percent component testing
- 25 percent system testing

نمونه ای از برخی قواعد اکتشافی

- خزش نیازهای کاربران با نرخ متوسط ۲ درصد در ماه از طراحی تا کدنویسی افزایش خواهد یافت
- نقاط عملکردی به توان ۱.۲۵ پتانسیل تقریبی نقص را در پروژه های نرم افزاری جدید پیش بینی میکند
- هر بازرسی رسمی طراحی ۶۵ درصد از اشکالات موجود را پیدا و حذف می کند.
- برنامه نویسان تعمیر و نگهداری می توانند در هر ماه هشت باگ را تعمیر کنند.
- نقاط عملکردی که تا توان ۰.۴ افزایش یافته اند، زمانبندی تقربی را پیش بینی می کنند
- نقاط عملکردی تقسیم بر ۱۵۰ تعداد تقریبی پرسنل مورد نیاز را پیش بینی می کند
- نقاط عملکردی تقسیم بر ۷۵۰ تعداد تقریبی کارکنان تعمیر و نگهداری مورد نیاز برای به روز نگه داشتن برنامه را پیش بینی می کند.

accurate. As Garmus and Herron point out (Garmus and Herron 1996):

> ırate estimating is a function of applying a process and recognizing that effort must be expended in creating a baseline of experience that will allow for increased accuracy of that process. Estimating does not require a crystal ball; it simply requires commitment (142).

بهترین روش ارزیابی چیست؟

- هیچ روش بهترینی وجود ندارد!
  - استفاده از تجربه
  - استفاده از چند روش
  - اجتناب از رژه مرگ
  - کار طولانی و ۲۴ ساعته راه حل مسایل طولانی نیست!

If you can’t measure it,
you can’t improve it.

Peter Drucker

راهنمای کار با نرم افزار مدیریت پروژه

- راهکار توسعه ساختار شکست کار به صورت تیمی
- کیفیت تقسیم بندی
- کیفیت تخمین زمان