UNIVERSITÄT
BERN

Project Management

مدیریت پروژه

دکتر احمد تقی نژاد

سرفصل مطالب

UNIVERSITÄT BERN

- مدیریت ریسک
- محدوده و تخمین
- برنامه ریزی و زمانبندی
- مقابله با تاخیرها
- استخدام ,هدایت ,کار تیمی

منابع

Sources

- *Software Engineering*, I. Sommerville, 7th Edn., 2004.
- *Software Engineering — A Practitioner's Approach*, R. Pressman, Mc-Graw Hill, 5th Edn., 2001.

Recommended Reading

- *The Mythical Man-Month*, F. Brooks, Addison-Wesley, 1975
- *Object Lessons*, T. Love, SIGS Books, 1993
- *Peopleware, Productive Projects and Teams* (2nd edition), Tom DeMarco and Timothy Lister, Dorset House, 1999.
- *Succeeding with Objects: Decision Frameworks for Project Management*, A. Goldberg and K. Rubin, Addison-Wesley, 1995
- *Extreme Programming Explained: Embrace Change*, Kent Beck, Addison Wesley, 1999

سرفصل مطالب

- **مدیریت ریسک**
- محدوده و تخمین
- برنامه ریزی و زمانبندی
- مقابله با تاخیرها
- استخدام ,هدایت ,کار تیمی

۱-«به مدیرعامل‌مان گفتم که طراحی به زودی آماده می‌شود. حسابی خوشحال شد.»

۲- ولی من به شما گفته بودم که تکمیل آن شش ماه طول می‌کشد.

OOH.

۳-باید بهشون بگی

۴-خب، دیگر خیلی دیر شده

۵-او قبلاً بیانیه‌ی خبری صادر کرده. باید طراحی را ظرف یک ماه تمام کنید.

۶-«تنها راه این است که نقص‌های عمده‌ی طراحی را بپذیریم که باعث نابودی خط تولید میلیارد دلاری می‌شود.»

۷-«مشکلی نیست. گزینه‌های سهام من آن‌قدر بی‌ارزش شده که دیگر اهمیتی ندارد.»

۸-«تمام مشکلات را گردن شرکت چینی تولیدکننده‌ی محصولات‌مان می‌اندازم.»

۹-«در نهایت، مقصر اصلی مدیرعامل است که مشوق‌های مناسبی برایم فراهم نکرده.»

http://en.wikipedia.org/wiki/Fair_use

مدیریت پروژه چیه؟

چالش پروژه = *تحویل درسر وقت و **با بودجه تعیین شده***

مدیریت پروژه = *برنامه ریزی برای کار و انجام کار با برنامه*

وظایف مدیریتی

- *برنامه ریزی:* برآورد و برنامه ریزی منابع
- *سازمان:* چه کسی چه می کند
- *کارکنان:* استخدام و انگیزش پرسنل
- *هدایت:* اطمینان از اینکه تیم به عنوان یک کل عمل می کند
- *نظارت (کنترل):* انحرافات برنامه + اقدامات اصلاحی را شناسایی کنید

مدیریت ریسک

اگر به‌طور فعالانه به ریسک‌ها حمله نکنید، آن‌ها به‌طور فعالانه به شما حمله خواهند کرد.

—تام گیلب

ریسک‌های پروژه

بودجه، برنامه‌ریزی، منابع، اندازه، نیروی انسانی، روحیه و انگیزه...

ریسک‌های فنی

فناوری پیاده‌سازی، اعتبارسنجی، نگهداری و تعمیرات...

ریسک‌های کسب‌وکار

بازار، فروش، مدیریت، تعهد...

مدیریت ریسک...

UNIVERSITÄT BERN

مدیریت باید:

- شناسایی ریسک در اسرع وقت
- ارزیابی اینکه آیا ریسک قابل قبول هستند
- اقدامات لازم را برای کاهش و مدیریت ریسک انجام دهید
  - برای مثال، نمونه سازی، ...
- نظارت بر ریسک ها در طول پروژه

تکنیک های مدیریت ریسک

تکنیک مدیریت ریسک	موارد ریسکی
کارکنان با استعداد بالا؛ تیم سازی؛ آموزش متقابل؛ پیش برنامه ریزی افراد کلیدی	کمبود کارکنان
هزینه دقیق ومنابع مختلف و برآورد زمانبندی؛ توسعه تکمیلی؛ استفاده مجدد از نرم افزار؛ کم کردن نیازمندها	*زمانبندی و بودجه غیر واقع بینانه*
نظرات کاربران، نمونه سازی، راهنمای کاربران اولیه	توسعه توابع نرم افزاری اشتباه

تکنیک های مدیریت ریسک...

UNIVERSITÄT BERN

تکنیک مدیریت ریسک	موارد ریسکی
آستانه تغییر بالا؛ پنهان سازی اطلاعات(پیشبینی تغییرات و پنهان کردن اثرآنها روی با ماژوله کردن)؛ توسعه تکمیلی(قادرساختن برای انجام تغییرات در ورژن بعدی)	جریان تغییرات نیازمندی ها
مدل benchmarking; شبیه سازی; *tuning ;بکارگیری ابزار مناسب* ;سازی	ایرادات کارایی realtime

نقشه‌راه

- مدیریت ریست
- **محدوده و تخمین**
- برنامه ریزی و زمانبندی
- مقابله با تاخیرها
- استخدام ,هدایت ,کار تیمی

Focus on Scope

*برای چندین دهه برنامه نویسان گله مند بوده اند: "**مشتریان نمی توانند** ما را **به آنچه می خواهند بگویند**. وقتی آنها را به آنچه می گویند می خواهیم، آن ها را دوست ندارند. این حقیقت مطلق توسعه نرم افزار است. الزامات در ابتدا هرگز روشن نیست. مشتریان هرگز نمی توانند دقیقا همان چیزی را که می خواهند بگویند.*

- **Kent** Beck

داستان: دامنه و اهداف

Myth

"یک بیانیه کلی از اهداف برای شروع برنامه نویسی کافی است."

Reality

تعریف ضعیف اولیه، علت اصلی شکست پروژه است.

Scope and Objectives

برای برنامه ریزی، باید دامنه و اهداف روشن را تعیین کنید

- **اهداف**: تعیین مقصد کلی پروژه، نه نحوه دستیابی به آنها.
- **محدوده**: مشخص کننده توابع اولیه ای است که نرم افزار برای انجام آن ها را تعیین می کند و این عملکردها را به شیوه ای کمی محدود می کند.

اهداف باید واقع بینانه و قابل سنجش باشند

- محدودیت ها، کارایی، قابلیت اطمینان باید صریحا بیان شود
- مشتری باید اولویت ها را تعیین کند

استراتژی های تخمین

These strategies are simple but risky:

با کارشناسان مشورت کنید و تخمین ها را مقایسه کنید ارزان، اما غیر قابل اعتماد	***قضاوت کارشناس***
مقایسه با سایر پروژه ها در یک دامنه کاربرد مشابه محدودیت پذیری: باید پروژه مشابه باشد	***برآورد توسط قیاس***
این پروژه هرچقدر که منابع در دسترس هستند هزینه دارد مزایا: بدون مصرف بیش از حد معایب: سیستم معمولا ناتمام است	***قانون پاریکنسون***
پروژه به اندازه ای که مشتری میتواند روی آن هزینه کند می ارزد.	***قیمت گذاری برای پیروزی***

تکنیک‌های تخمین

تجزیه "و" مدل سازی الگوریتمی هزینه "با یکدیگر استفاده می شوند"

برآورد هزینه اجزا + یکپارچه سازی و ادغام تخمین بالا یا پایین ممکنه هزینه های حل مسائل سطح پایین سخت فنی را ناچیز بگیرد	***تجزیه***
رویکرد فرمولی مبتنی بر اطلاعات هزینه تاریخی است که عموما بر اساس اندازه نرم افزار است	***مدل سازی الگورتیمی هزینه***

تکنیک‌های تخمین

تخمین تجزیه‌ای(Decomposition Estimation)

تعریف:

در این روش، پروژه به اجزای کوچکتر (ماژول‌ها، توابع، زیرفرایندها) تقسیم می‌شود. ابتدا هزینه هر جزء به صورت مستقل برآورد می‌گردد و سپس مجموع هزینه‌ها به همراه هزینه‌های اضافی یکپارچه‌سازی و ادغام محاسبه می‌شود.

فرض کنید پروژه شامل n مؤلفه باشد. هزینه کل تخمین‌زده شده به صورت زیر بیان می‌شود:

$$Cost_{Integration} + \sum_{i=1}^{n} Cost_{component} = Cost_{total}$$

تکنیک‌های تخمین: تجزیه‌ای

یک پروژه شامل سه مؤلفه‌ی اصلی است:

- مؤلفه A با هزینه تخمینی ۲۰۰ نفر-ساعت
- مؤلفه B با هزینه تخمینی ۱۵۰ نفر-ساعت
- مؤلفه C با هزینه تخمینی ۲۵۰ نفر-ساعت
- هزینه‌ی یکپارچه‌سازی و تست ۱۰۰ نفر-ساعت تخمین زده شده است.

پس:

$$\text{CostTotal} = 200 + 150 + 250 + 100 = 700 \text{ نفر} - \text{ساعت}$$

تکنیک‌های تخمین: تخمین بالا به پایین و پایین به بالا

- **تخمین بالا به پایین**: از دید کلان به پروژه نگاه می‌شود و ابتدا هزینه کلی تخمین زده می‌شود و سپس به بخش‌های فرعی تخصیص داده می‌شود.
  - تخمین بالا به پایین ممکن است هزینه‌ی مشکلات فنی سطح پایین (مانند رفع اشکالات پیچیده یا مشکلات عملکردی) را ناچیز در نظر بگیرد.

$$\text{Estimated Total Cost} = \text{High-Level Estimate} \times (1 + \text{Risk Adjustment Factor})$$

- **تخمین پایین به بالا**: از جزئیات کوچک پروژه شروع می‌شود و سپس هزینه‌ها برای رسیدن به تخمین کلی جمع می‌شوند.

$$\text{Estimated Total Cost} = \sum_{i=1}^{n} DetailedTask$$

تکنیک‌های تخمین:تخمین بالا به پایین و پایین به بالا

مثال:

- تخمین اولیه برای یک پروژه: ۱۰۰۰ نفر-ساعت (بالا به پایین)
- ریسک‌های شناسایی شده: ۱۰٪
- پس:
- Adjusted Cost=1000×(1+0.10)=1100 نفر-ساعت

در مقابل، در پایین به بالا:

- ۵۰ کار کوچک با میانگین ۲۰ نفر-ساعت تخمین زده شده:
- 50×20=1000personHour نفر

تکنیک تخمین: مدل‌سازی الگوریتمی هزینه (Algorithmic Cost Modeling)

این روش بر اساس داده‌های تاریخی پروژه‌های قبلی و روابط ریاضی بین متغیرهای کلیدی (مانند اندازه نرم‌افزار، پیچیدگی، و تجربه تیم) استوار است. معروف‌ترین مدل در این زمینه مدل COCOMO (Constructive Cost Model) است.

$$\text{Effort}=Size^{b} \times a$$

که:

- Effort هزینه به نفر-ماه است.
- Size اندازه‌ی نرم‌افزار بر حسب هزار خط کد (KLOC) است.
- a و b ضرایب تجربی هستند که بسته به نوع پروژه (ارزشیابی شده از پروژه‌های تاریخی) تعیین می‌شوند.

تکنیک تخمین: مدل‌سازی الگوریتمی هزینه (Algorithmic Cost Modeling)

مثال:

فرض کنیم پروژه‌ای ۵۰ KLOC حجم دارد و ضرایب برای نوع پروژه‌ی "نیمه-محافظه‌کارانه" به صورت a = 3.0 و b=1.12 باشند. آنگاه:

$$\text{Effort}=50^{1.12} \times 3 = 248.4 \text{ نفر ماه}$$

مقایسه تکنیک‌های تخمین

روش	مزیت	محدودیت
تجزیه‌ای	دقت در سطح مؤلفه	زمان‌بر بودن در پروژه‌های بزرگ
بالا به پایین/پایین به بالا	سرعت (بالا به پایین)، دقت (پایین به بالا)	ریسک نادیده گرفتن جزئیات (بالا به پایین)
الگوریتمی	استانداردسازی و مقایسه‌پذیری	نیاز به داده‌های تاریخی معتبر

نقشه‌راه

- مدیریت ریست
- محدوده و تخمین
- **برنامه ریزی و زمانبندی**
- مقابله با تاخیرها
- استخدام ,هدایت ,کار تیمی

http://en.wikipedia.org/wiki/Fair_use

برنامه‌ریزی و زمان‌بندی

برنامه ریزی خوب بستگی به شهود و تجربه مدیر پروژه دارد!

- تقسیم پروژه به وظایف

  وظایف به کارهای زیر و غیره.

- تخمین زمان برای هر وظیفه
  - تعریف وظایف به اندازه ی کوچک که تخمین زمان مورد نیاز آن واقعی باشد .
- وظایف مهم باید با یک نقطه عطف به پایان برسد..
  - نقطه عطف پروژه جایی است که گزارش پیشرفت پروژه به مدیریت ارسال میشود
  - نقطه عطف غیر مبهم و واضح یک ضرورت است!

("80% coding finished" is a meaningless statement)

  - *نظارت بر پیشرفت کار توسط نقطه عطف*

برنامه‌ریزی و زمان‌بندی...

- **وابستگی** ها را در پروژه تعریف کنید

  Total time depends on longest (= critical) path in — activity graph

  —کم کردن وابستگی وظایف برای کاهش تاخیر
- سازمان دهی وظایف بصورت همزمان برای بهینه کردن استفاده از نیروی کار

برنامه ریزی تکراری میشود

⇒ نظارت و تجدید نظر برنامه ریزی در طول پروژه انجام میشود

افسانه: تحویل‌دادنی‌ها و نقاط عطف

Myth

"تنها چیزی که نیازه تا تحویل داده بشه، برنامه اجرایی است."

Reality

سندسازی همه جنبه های پروژه لازم است برای انجام نگه داری

تحویل‌دادنی‌ها و نقاط عطف

مفاد تحویلی پروژه چیزهایی هست که باید به مشتری تحویل داده شود

> برای مثال

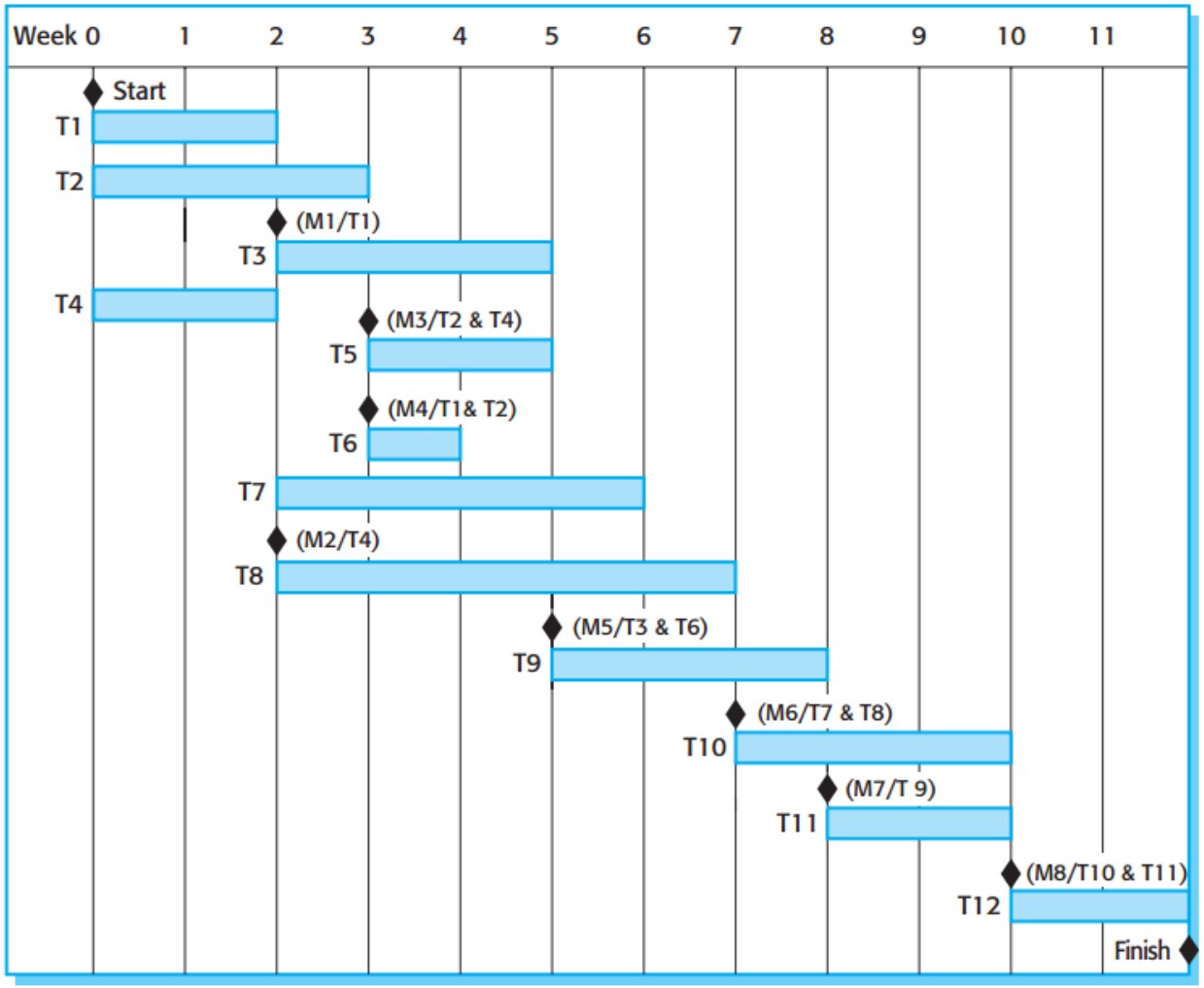
- سند اولیه مورد نیاز
- نمونه اولیه UI
- مشخصات معماری

مثال: مدت زمان وظایف و وابستگی‌ها

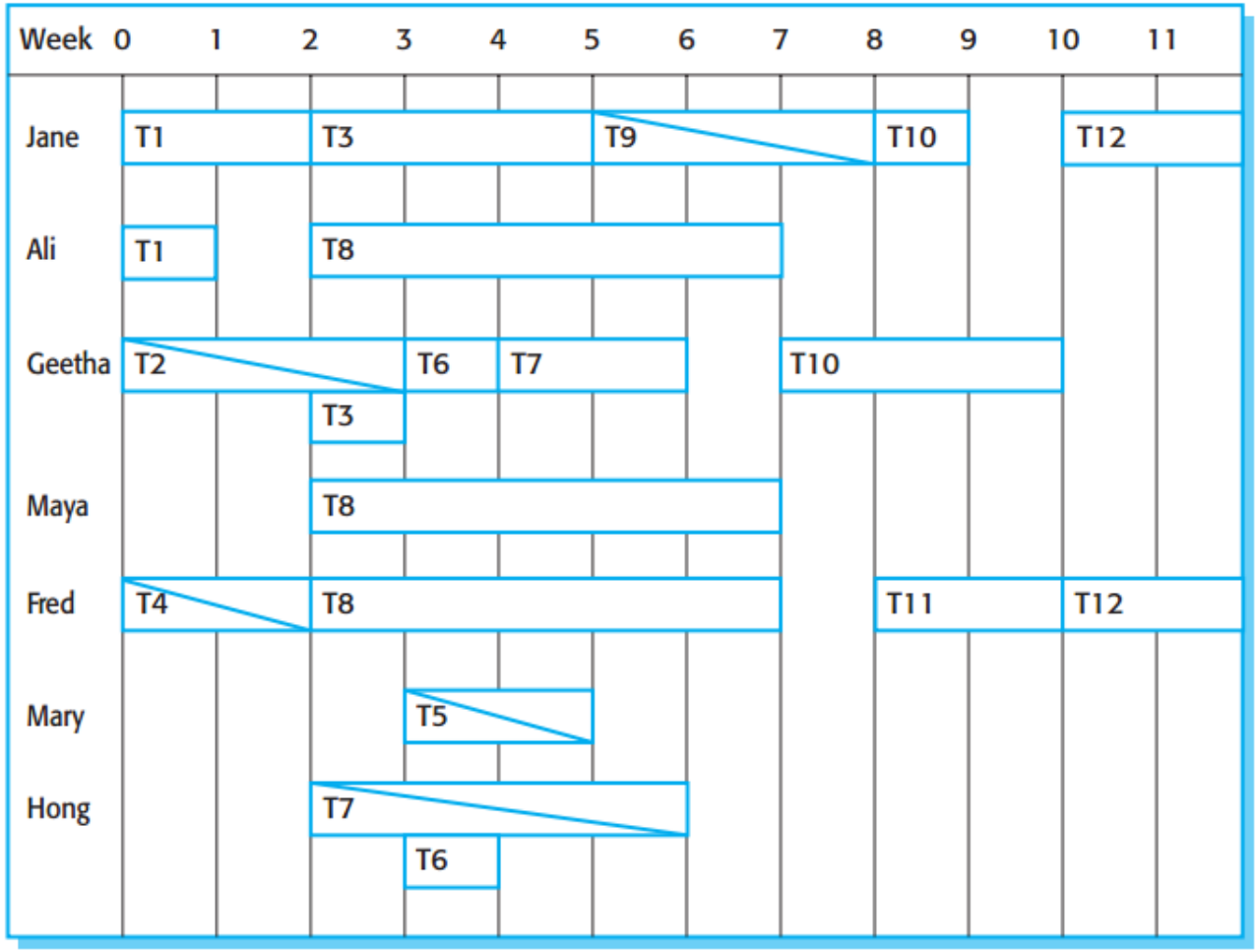
Task	Effort (person-days)	Duration (days)	Dependencies
T1	15	10	
T2	8	15	
T3	20	15	T1 (M1)
T4	5	10	
T5	5	10	T2, T4 (M3)
T6	10	5	T1, T2 (M4)
T7	25	20	T1 (M1)
T8	75	25	T4 (M2)
T9	10	15	T3, T6 (M5)
T10	20	15	T7, T8 (M6)
T11	10	10	T9 (M7)
T12	20	10	T10, T11 (M8)

What is the minimum total duration of this project?

نمودار میله‌ای فعالیت



نمودار گانت: نمودار تخصیص کارکنان



diagonal line crossing the bar a part time task

نقشه‌راه

- مدیریت ریست
- محدوده و تخمین
- برنامه ریزی و زمانبندی
- **مقابله با تاخیرها**
- استخدام ,هدایت ,کار تیمی

Myth: Delays

شایعه

"اگر از برنامه عقب بیفتیم ، می توانیم برنامه نویسان بیشتری اضافه کنیم و عقب نمانیم."

واقعیت

افزودن افراد بیشتر به طور معمول سرعت پروژه را کاهش می دهد.

مشکلات زمانبندی

- برآورد سختی مشکلات و هزینه توسعه یک مسئله دشوار است
- بهره وری با تعداد افراد کار بر روی یک وظیفه متناسب نیست
- اضافه کردن افراد به پروژه به تاخیر خورده باعث تاخیر بیشتر میشود به علت سربار ارتباطی
- *اتفاقات غیر منتظره همیشه رخ میدهند، همیشه احتمال این مسائل باید داده شود*
- کوتاه کردن از آزمون و بررسی دوباره در زمان بحران
- *کار کردن درطول شب یک راه حل کوتاه مدت است.*

برنامه ریزی تحت عدم اطمینان

- به وضوح مشخص کنید که چه می دانید و نمی دانید
- به وضوح مشخص کنید که چه کار خواهید کرد برای کاستن عوامل ناشناخته
- اطمینان حاصل کنید که نقاط عطف اولیه حتی اگر دوباره برنامه ریزی شود سر وقت باشد

مقابله با تاخیرها

نقاط محتمل تاخیرها را کشف کنید

به این طریق فرصت بیشتری برای بازیابی دارید ...

...

مقابله با تاخیرها

نحوه ریکاوری:

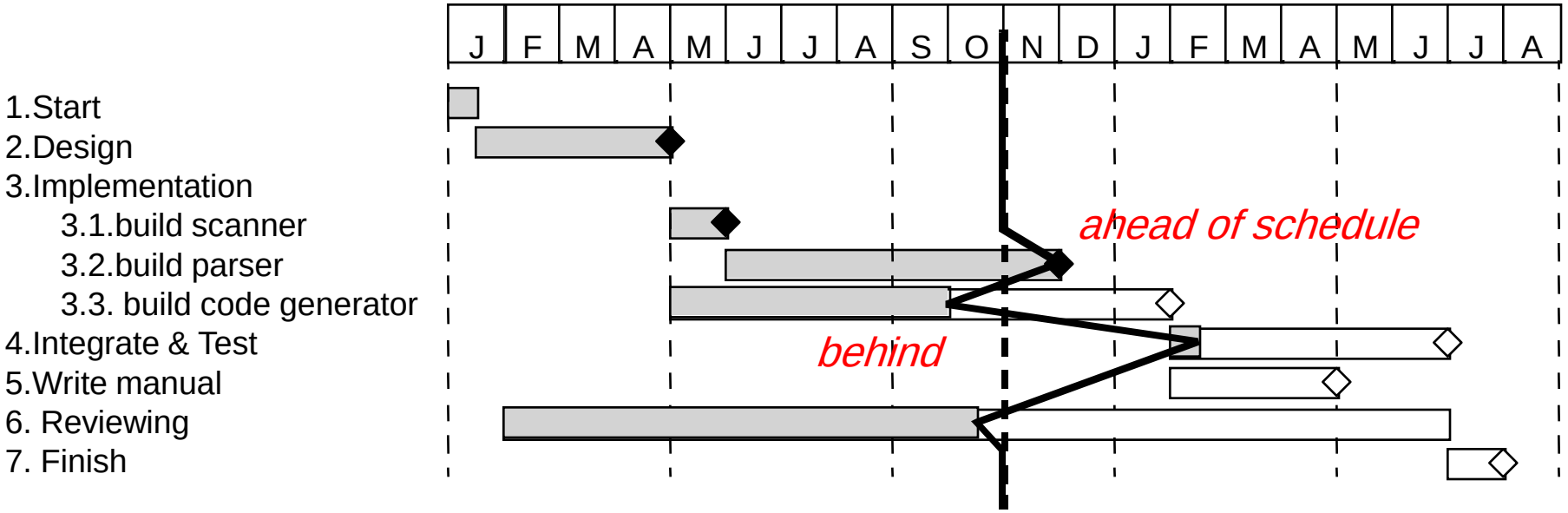
ترکیب این سه عمل است

- اضافه کردن کارمندان ارشد به وظایف مشخص شده
  - خارج از مسیر بجرانی برای جلوگیری از سربار ارتباط
- اولویت بندی نیازمندی ها و تحویل به صورت تدریجی
  - مهمترین عملکردها رو سر وقت تحویل دهید
  - تست کردن یک اولویت باقی می ماند (even if customer disagrees)
- تمدید مهلت انجام

Gantt Chart: Slip Line

Visualize slippage متصور کردن تاخیر

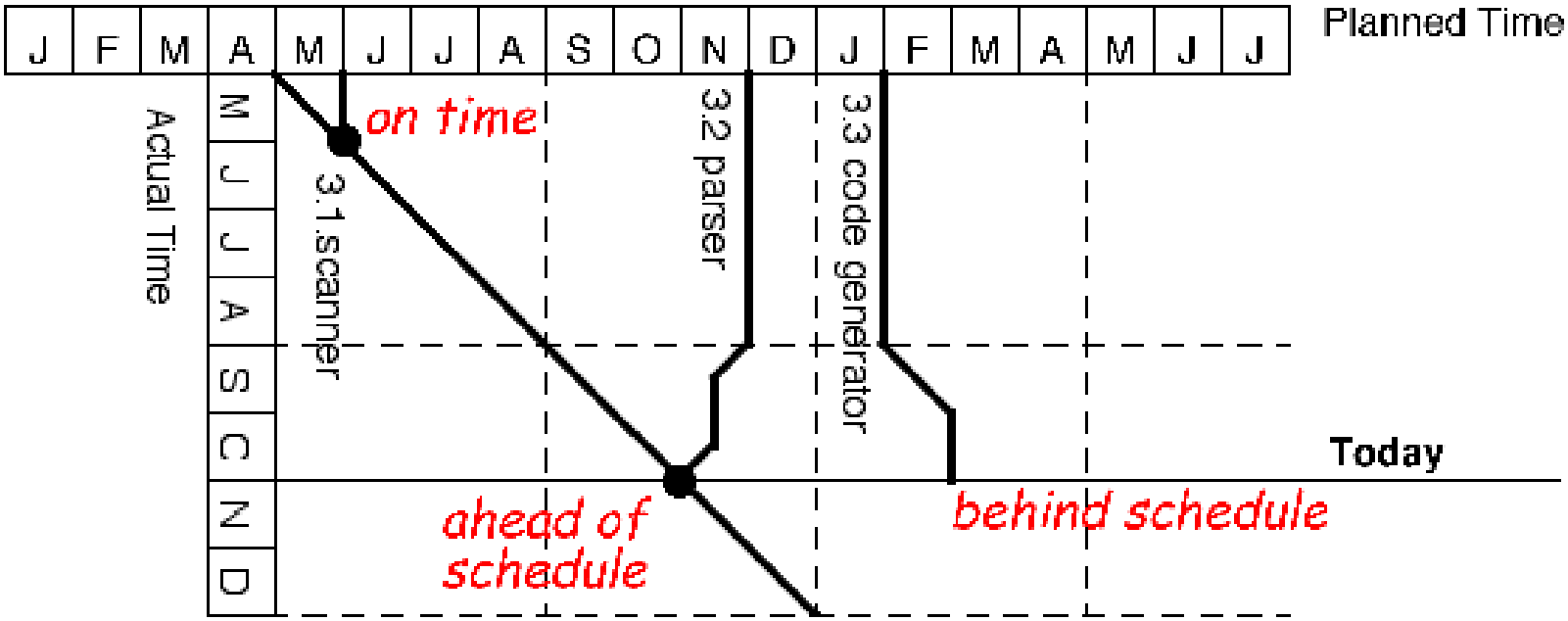
- Shade time line = portion of task completed
- Draw a slip line at current date, connecting endpoints of the shaded areas
  - bending to the right = ahead of schedule
  - to the left = behind schedule



Timeline Chart

Visualise slippage evolution متجسم کردن تکامل تاخیر

> bullets at the end of a line represent completed tasks



نقشه‌راه

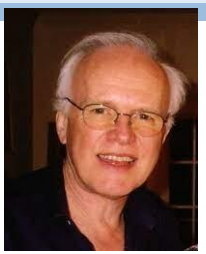
- مدیریت ریست
- محدوده و تخمین
- برنامه ریزی و زمانبندی
- مقابله با تاخیرها
- **استخدام ,هدایت ,کار تیمی**

Software Teams

سازماندهی تیمی

- تیم ها باید نسبتا کم باشند (کمتر از ۸)
  - حد اقل کردن سربار ارتباطی
  - استاندارد کیفیت تیم را می توان توسعه داد
  - اعضا میتوانند با یکدیگر همکاری نزدیکی داشته باشند
  - برنامه ها به عنوان اموال تیم ("برنامه نویسی بی نظیر") در نظر گرفته می شود
- پروژه های بزرگ را به چند پروژه کوچکتر برسانید
- تیم های کوچک ممکن است به شکل غیر رسمی و دموکراتیک سازماندهی شوند

تیم گردانی، هدایت تیم



مدیران در خدمت تیم خود هستند

- مدیران اطمینان می دهند که تیم اطلاعات و منابع لازم را در اختیار دارد

"وظیفه مدیر این نیست که مردم کار کند ، بلکه این امکان را برای مردم فراهم می کند که کار کنند"

— Tom DeMarco

مسئولیت نیازمند اعتبار است

- مدیران مسئولیت میدهند

—به تیم خود اعتماد کنید تا آنها هم به شما اعتماد کنند.

تیم گردانی، هدایت تیم

مدیران مدیریت می کنند

- مدیران نمی توانند وظایف در مسیر بحرانی را انجام دهند
  - مخصوصاً برای مدیران فنی دشوار است!

توسعه دهندگان مهلت ها را کنترل می کنند

- یک مدیر نمی تواند مهلتی را که توسعه دهندگان با آن موافقت نکرده اند تأمین کند

UNIVERSITÄT BERN

آنچه باید بدانید!

- چگونه نمونه سازی اولیه می تواند در کاهش خطر در یک پروژه کمک کند؟
- نقاط عطف چیست ، و چرا آنها مهم هستند؟
- از شبکه فعالیت می توانید چه چیزی یاد بگیرید؟ جدول زمانی فعالیت؟
- چرا تیم های برنامه نویسی نباید بیشتر از حدود ۸ عضو داشته باشند؟

The End

Question?