S M T W T F S

20/02/2024

SUBJECT

Software's size — LOC (Lines of Code)
— Function Point (khó tính, LOC phổ biến hơn)

2 cách chia SW:
SW System
SW Sys vs Application

- SW System: Operation System for example. ← về
+ ← về phần cứng, phải hiểu phần cứng, thường xuyên lập trình
ngôn ngữ bậc thấp.

- Generic vs Custom SW ("may đo")
(large size) → main goals

SW — Documentation — Thiết kế, Kỹ thuật, HDSD.
— Data Structure.

Đây đủ các yếu tố CSHT → Tạo ra SW project.

IC → ——— → ——— IT project.

Ở VN hiện nay, mà ngân hàng, đều là xây IT project.

Scanned with CamScanner

1) VD 10 tiêu chí ISO 9126.

2) Đạo đức nghề nghiệp explain 10 điều cái.

S M T W T F S

/ /

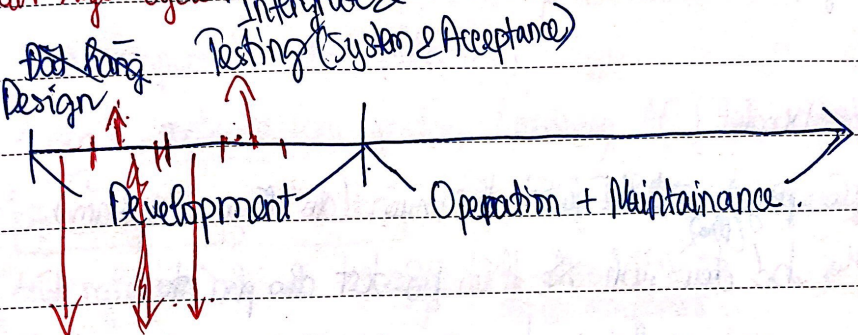
SUBJECT

Evaluate a software product

- Speculation → So xem SP đúng vs yêu cầu ~~và~~ đặc tả không?
  - + Tất cả các yêu cầu đc chấp nhận → CLC.
  - → Tiêu chí (có cách đo đạc không nhất thiết đầu)

-

SW life-cycle



↳ Biến design thành product.

Số lần ope + maintainance tỷ lệ thuận với chất lượng.

Scanned with CamScanner

S M T W T F S

/ / SUBJECT

V- Model (Verification & Validation Model)

Mô hình phát triển phần mềm

↳ Cách phát triển phần mềm hiệu quả. điển hình 1 cách áp dụng 1 tập ứng dụng với định

Mỗi cty có cách làm riêng → Tổng kết lại → đưa ra mô hình chung phát triển ~~chưa~~ có hiệu quả.

V- Model ( )

Giải quyết vấn đề kiểm thử muộn (do chậm ra sản phẩm).

Mỗi bước (pha) đều sinh ~~the~~ thêm bộ test cho pha đó.

— , sinh test sớm, chạy bộ test sau này

⇒ Kiểm thử mịn hơn

Bây dung khác nó là dùng V-Model.

Scanned with CamScanner

S M T W T F S

/ /

SUBJECT

Component-based Model

Tổng quan - ý tưởng chính

- Là 1 mô hình theo chiến lược reuse-oriented SE.
- Hướng đến phát triển ứng dụng lớn QC = cách SD lại các tphần phần mềm đã đc SD trước đây (đã đc vận hành trong các hệ thống hiện tại) - well defined component.

Đ/n component (tphần phần mềm):

+ 1 đơn vị software (gói - package / module) đóng gói 1 tập các dịch vụ liên quan đến nhau (dùng chung dữ liệu, cùng miền quy trình nghiệp vụ). cung cấp các service cần thiết cho các component.
+ Mới tphần đều có tên, giao diện và code.

- Vẫn theo quy trình đặc tả, ptích, thiết kế, nhưng 1 phần des và hầu hết các cài đặt là kết hợp từ các tphần sẵn.

Các bước:

- Đọc thu thập, ptích đặc tả yêu cầu.
- Ptích các tphần (yêu cầu nào hợp vs tphần nào

TOTO

Scanned with CamScanner

S M T W T F S

/ / SUBJECT

- Sửa đổi yêu cầu.
- Thiết kế hệ thống vs SP return.
- Phần Development and Integration (lắp thêm component việc tích hợp).
- Thẩm định SP.

: thêm / thay thế component tương ứng vs tính năng

⊛ Ưu điểm:

- SP đc đảm bảo chất lượng rất cao (các phần mềm đã đc đảm bảo chương khá tốt).
- Linh hoạt: Ra khi muốn add/replace feature ⇒ add/replace component tương ứng vs tính năng.
- Tiết kiệm thời gian
- Dễ phát hiện ra lỗi do các component độc lập.

⊛ Nhược điểm:

- Tìm component phù hợp khá khó khăn (tìm kết các tiêu chí)
- Việc đảm bảo tính nhất quán giữa các phần ≠ nhau của tập hợp các component có là 1 khó khăn

TOT

Scanned with CamScanner

S M T W T F S

/ / SUBJECT

⊛ Usage:

- T/chất phức tạp cao
- SR phù đối hỏi phải linh hoạt, đc đảm bảo chất lượng rất cao.

⊛ Các đặc điểm của component:

- Có thể kết hợp đc: Tương tác giữa các component thể hiện qua các giao diện của chúng (công khai, rõ ràng)
- Triển khai đc: Hoạt động độc lập trên nền tảng cung cấp implementation cho các component model.
- Tuỳ biến đầy đủ → user's decision.
- Độc lập: Các component có thể đc triển khai, kết hợp mà k° cần phụ ∈ vào component ≠.
- Tiêu chuẩn: Tuân thủ tiêu chuẩn của 1 component model (giao diện data, document, implementation

Scanned with CamScanner

S M T W T F S

/ / SUBJECT

RUP – Rational Unified Process

- Hay đc SD trong các dự án lớn ~~(cho chi~~
- Xây dựng CNTT cho Chính phủ thường SD RUP.
- Gắn liền vs phiên hướng đối tượng, UML

RUP = WAY + tiến hoá + SD lại thành phần

Scrum

1) Tổng quan – ý tưởng chính

- Phiên thuần theo p² luận Agile
- Chia dự án thành các phần nhỏ (sprint) cho dễ quản lý.

quan trọng, cung cấp cấu trúc phiên SW lặp lại, tăng dần!

+ chu trình có GIỚI HẠN để hoàn thành 1 công việc cụ thể (~1 tháng)

+ đc ấn định tg cụ thể (dù hoàn thành hay chưa)

TOTO

Scanned with CamScanner

S M T W T F S

/ /

SUBJECT

bao gồm tính năng đc ưu tiên của khách hàng.

① Các vai trò:

- Product Owner: ng quản lý product backlog
- Scrum Master: ng quản lý, hướng dẫn team thực hiện Scrum
- Development Team: team phát triển SW.

→ quyết định chính làm gì.

2) Các bước: Product owner, stakeholders đưa yêu cầu sản phẩm

- Scrum planning (đầu mỗi sprint): Chọn ra ít tính năng ưu tiên từ product backlog có thể hoàn thành đc để phát triển trong sprint → Sprint backlog

↳ Dev Team dựa trên để làm việc.

- Daily Scrum: Cuộc họp đc tổ chức HẰNG NGÀY

+ Các trên cần trả lời đc:
- đã làm gì hôm qua
- sẽ làm gì hôm nay
- vấn đề gặp phải → điều chỉnh

- Sprint Review (cuối mỗi sprint), Trình bày vs KH ng gì Dev Team làm đc, nhận phản hồi để thực hiện trong sprint kế tiếp.

TOTO

Scanned with CamScanner

S M T W T F S

/ / SUBJECT

⊗ Scrum quan tâm sản phẩm cuối mỗi sprint: phần đc tích hợp, hệ thống, đc kiểm thử đầy đủ, hoàn thiện tài liệu cho ng dùng, chuyển giao etc.

- Sprint Retrospective: Bế mạc cuối cùng.
  + Kết luận sprint này diễn ra tốt ở gì
    ⌊ ở điểm tốt / chưa tốt / cần cải thiện thêm.

3/ Ưu điểm:

- Chất lượng SP tăng ↑ Do mỗi sprint tập trung vào 1 mục tiêu as thế → Dev Team tập trung hoàn thành các mục quan trọng.
  + Chia nhỏ dự án → giảm sai sót.
  + Nhận đc phản hồi từ KH sau mỗi sprint → Tăng chất lượng.

- Tăng sự hài lòng của KH: Do KH đc tgia xuyên suốt quá trình sản xuất, phản hồi → tăng SP đáp ứng đc ñ nhu cầu của KH.

TOTO

Scanned with CamScanner

S M T W T F S

/ /

SUBJECT

- Tăng năng suất. Tgian của sprint cố định ⇒ Team sẽ làm ñ việc hơn dưới áp lực tgian.

* Nhược điểm:

- Khó làm quen vs Scrum trong tgian đầu do Scrum k° phải mô hình đc mô tả đầy đủ
- Team phải cái gì cx cần biết (tất cả các khâu) cũng có kinh nghiệm do Scrum gồm các giai đoạn làm việc cường độ cao, đòi hỏi việc phải hoàn thành trong tgian ngắn.

* Tình huống SD:

- Dự án yêu cầu CLC, phức tạp, đòi hỏi luôn phân đc ñ phản ñ thay đổi, biến động.
- Dev Team ñ kinh nghiệm, có khả năng tự quản lý kết hợp thử thách năng.

TOTO

Scanned with CamScanner

~~Qua~~

S M T W T F S

05 / 05 / 2024 SUBJECT

RAD Quản lý nhóm n → GT QLDA.

(1 trong những việc khó nhất trong SW Dev, đòi hỏi chuyên gia có kinh nghiệm).

RAD $SW = S_1 \| S_2 \| \dots \| S_n$.

⊛ Làm thế nào đảm bảo tính nhất quán giữa n team?

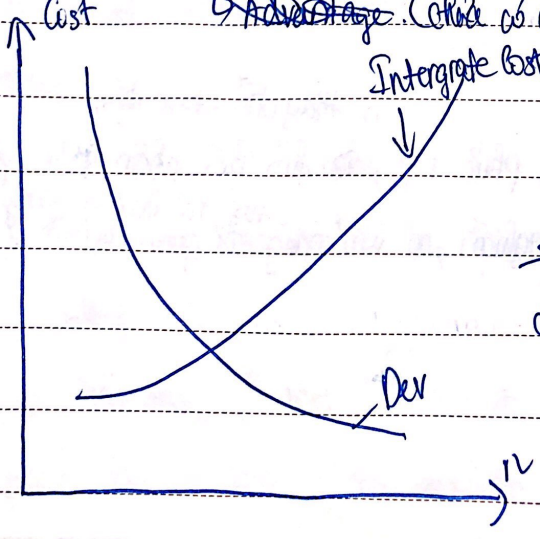
→ Phải có 1 kiến trúc sư đại thống

TL_1 TL_2 ... TL_n

Ph4

$$\boxed{Cost(S) \geq Cost(S_1) + C(S_2) + \dots + C(S_n)}$$

↳ ~~Advantage~~ (chưa có chi phí tích hợp)



→ Việc tìm n tối ưu cần dựa vào kinh nghiệm thực tiễn.

Scanned with CamScanner

Hello

S M T W T F S

/ /

SUBJECT

CBSD → SOA.

- Áp dụng hệ thống reuse-oriented
- Việc kết hợp các component phải đáp ứng "plug-and-play".

CBSD VS SOA (TRÊN CÙNG MÔI TRƯỜNG)

SOA là 1 phiên bản đặc biệt của CBSD (các component online k° quan tâm đến công nghệ)

→ Đưa đến quy luật hợp tác các bên đều có lợi.

→ Phát triển PM SW tg ngắn (k° chắc chắn về chi phí)

TDD (viết và nhớ).

User stories —(informal language)→ Test scripts.

↳ Test chức năng, UI.

RUP lặp đi lặp lại tăng dần → hệ thống, Spiral

Scanned with CamScanner

S M T W T F S

/ /

SUBJECT

System Modeling

① ~~Class~~ Một số khái niệm cơ bản:

- System = Software + $\underbrace{\text{devices + humans + ...}}_{\text{những bên ngoài}}$
- Mô hình hoá hệ thống là đơn giản hoá hệ thống trong thế giới thực.

② Mục đích.

- Hiểu tổng thể hệ thống + giao tiếp các bên liên quan
- Mô hình ~~hệ~~ system-as-is
- ——— system-to-be (phục vụ cho phân tích, thiết kế, cài đặt)

I/ Class Diagram

① Class Diagram ~~đc~~ hiểu là trung tâm của UML

Đa dạng

Scanned with CamScanner

S M T W T F S

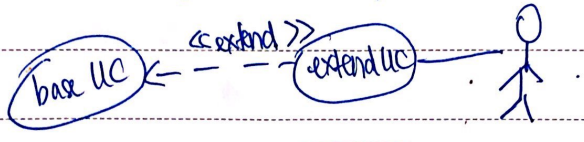
/ / SUBJECT

Collaboration <=> Sequence

Dù là diagram nào thì cx gồm các thường và mối quan hệ giữa chúng

Xđịnh actor → Xđịnh usecase → "làm mịn" usecase

* Generalization: Tổng quát hoá usecase (khi mở rộng usecase gốc đc dùng cho 1 thằng ≠ → phụ thuộc kế)
* Included usecase phải đc included từ ≥ 2 usecase.
  ↳ Tách riêng ra để phục vụ thiết kế.
  → Included usecase k° phải 1 usecase độc lập. Actor k° thể tương tác trực tiếp vào đc.



Extended UC là 1 UC $\boxed{\text{độc lập}}$.

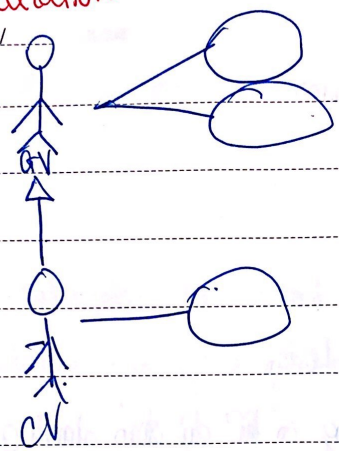
Scanned with CamScanner

S M T W T F S

Generalization

/ /

SUBJECT



→ CVHT sẽ có 3 usecase

Format: Slide số 4 (mô tả usecase)

Behavior: Thể hiện các methods thực hiện.

Usecase → Class Diagram từng use case → merge gộp thành Class Diagram hoàn chỉnh.

Object Diagram giúp thử validate Class Diagram

Scanned with CamScanner

S M T W T F S

/ / SUBJECT

Thiết kế kiến trúc

- Nghĩ chung
- Thiết kế
  - AD
  - DBase
  - UI | UX
  - Detailed Design

Tại sao làm thiết kế xong có thể dự đoán đánh giá các yêu cầu phi chức năng.

Nên tập thói quen code dựa trên bản thiết kế.

Thiết kế
- là quá trình sáng tạo
- Nghĩ xem nên làm như (How ?)
- Xem xét lại, chi tiết hoá.

⇒ Cần đủ chi tiết cho đội ngũ lập trình.

Thế

Thiết kế phần mềm là tạo mô hình cấu trúc phần mềm để ~~tuân thủ~~ đảm bảo ~~thực~~ thi phương tiện để truyền tải thông tin, đảm bảo chất lượng.

TOTO

Scanned with CamScanner

Mẫu hiểu thực.

S M T W T F S

/ /

SUBJECT

KH đa số chỉ quan tâm đến tài liệu đặc tả, k° quan tâm tài liệu thiết kế.

Module (subsystem, independent component, ...)
→ Modularity is a good strategy.

Thiết kế phải dựa trên phân tích
→ Đáp ứng đặc tả
ứng hoàn toàn.

Nhiều quan trọng biểu diễn

Làm mịn: Thêm / chi tiết đầu thông tin (phải Tự - hoàn).

Scanned with CamScanner

S M T W T F S

/ /

SUBJECT

Kiến trúc phần mềm.

K có đ/n chuẩn về KTPM, tuỳ theo quan đ' của từng tổ chức đó.

Richard et al.: KTPM là tập các quyết định nghề thiết kế của hệ thống.
Bản thiết kế xây dựng, triển khai PM.

L. Bass et al.: KTPM là tập các cấu trúc cần thiết để suy luận về hệ thống (gồm các phần, quan hệ giữa các phần, ∈ tính của 2 cái trên)

IEEE:

Tại sao KTPM lại quan trọng?

- Hạn chế, cho phép 1 ∈ tính của hệ thống.
- Suy luận / quản lý thay đổi.
- Thay đổi là điều tất yếu xảy ra.
- Phương tiện giao tiếp giữa các bên liên quan của hệ thống.

TOTO

Scanned with CamScanner

Chiến lược phi dịch vụ:

Chia nhỏ System thành ñ dịch vụ.

Thiết kế giao diện UI/UX.

Kịch bản nội dung thường đc thể hiện

Phân tích nội dung + Tuổi tác

+ Trình độ SD

+ Ngôn ngữ.

Cần quan tâm: _ Input Output.

_ Cách biểu diễn.

Đề dùng:

_ Tgian học. (Phải chia theo nhóm)

_ Tốc độ thực hiện v (Theo tgian)

_ Hài lòng chủ quan (Tường đối)

_ Tỷ lệ mắc lỗi nội dung (Số lỗi / đơn vị tgian)

_ Nhớ đc

Lỗi ntn (to 19)

TOTO

Scanned with CamScanner

S M T W T F S

/ / SUBJECT

Các ng Key Contents

1) 1 số giải pháp (có đặc tả, chưa có thiết kế) (mua or thuê & (thuê, mua sp, nguồn mở, thuê phát triển)? Ưu nhược điểm?

2) Implement (Từ chi tiết → sản phẩm)? Tsao lo gọi coding/programming? Nếu code phải tuân theo n ngôn ngữ gì?

Giải pháp phát triển sp phần

Giả sử A cần xây dựng sản phẩm X. Các giải pháp đặt ra:

- Tự làm
- SD nguồn mở
- Thuê X từ cty ABC
- Mua X ——— XYZ.
- Thuê B xây dựng, phát triển.

Tự làm:

a) Nhà đầu tư ... lo chuyên nghiệp thì mũ trở các cty lớn

Scanned with CamScanner

Implementing : Bản chất : Quá trình Thiết kế → sản phẩm
(chưa ~~test~~ test)

Coding tại là 1 cách Implementing

Tsao Implementing k° chỉ bao gồm Coding ?

Vì vs sản phẩm to thì k° thể code
(chỉ ok vs sản phẩm nhỏ)

TOTO

Scanned with CamScanner

Tại sao Implementing sti vs Unit Testing?

S M T W T F S

/ / SUBJECT

Chiến lược Implementing:

* Tăng cường Reuse (bán sử dụng, k° bán sở hữu)

~~Nên~~ tiện của các cty — xem thị trường có bán k°. (thuê / mua)

* Chỉ code khi k° reuse đc!

Tsao k° tự code, nên-reuse? ~~vì tốn~~ công (xem CBSD).

Tuân thủ code convention rule

Scanned with CamScanner

S M T W T F S

/ /

SUBJECT

⇒ Kiểm thử là quy trình kiểm chứng nhằm đánh giá + tăng cường chất lượng SW.

Kiểm thử
- là gì?
- Khi nào? vs SQA.
- ntn?

D → Implement + Unit Test → Code + Test + Báo cáo Test
(Coding)

Verification & Validation

⊛ Verification

In: D (đúng ý tác tả) → II

Out: II có ~~tc implement~~ + D (có thm D k°?)
- Y
- N + counterexample.

Mục đích: Tìm ra lỗi trong quá trình D → II.

⊛ Validation:

TOTO

Scanned with CamScanner

S M T W T F S

/ / SUBJECT

In : SRS - - - -> II.

Out : II ⊢ SRS ?

↳ Trừ lỗi thiết kế, implement.

=> Verification hơn Validation

Nếu Validation hơn thì k° thể xác định lỗi pha nào.

↳ Acceptance Testing là Validation.

V&V < Static / Dynamic.

① Static : Đọc code check ra lỗi (Chương 6 + 7).

② Dynamic : Chạy code

$\boxed{\text{SW Testing}}$ = Dynamic V&V.

SW Testing chỉ là 1 hoạt động SQA trong giai đoạn Implementing thành phần của Unit Testing.

SQA xuyên suốt hệ thống → cuối hoàng phần SW.

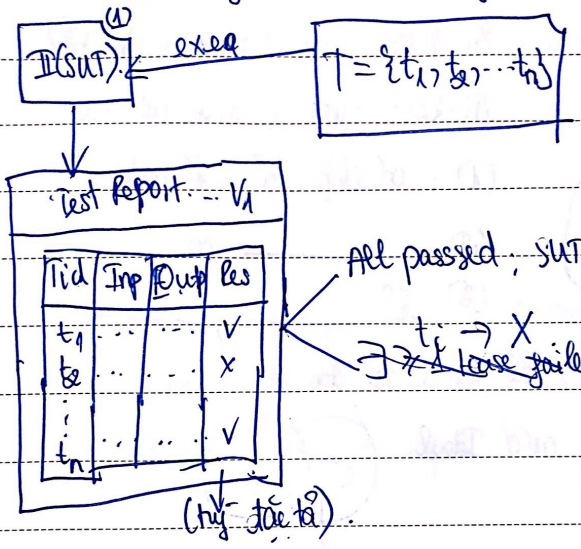
V - model là quy trình SQA.

Scanned with CamScanner

S M T W T F S

/ / SUBJECT

Bức tranh tổng thể SW Testing:



Testing → Debug → Π(SUT) v2

(phát hiện lỗi)

SW Testing < Sinh Test T / Chạy Test T → JUnit l° phải công cụ testing mà là công cụ chạy bộ test.

hỗ trợ

reuse → đỡ tốn công.

Scanned with CamScanner

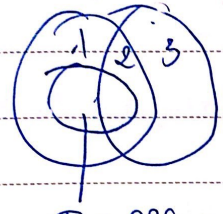
S M T W T F S

/ / SUBJECT

Sinh T như nào? (Generate / Design Test)

2 chiến lược: Black box Test (sinh từ đặc tả)

Whitebox Test (sinh từ code)

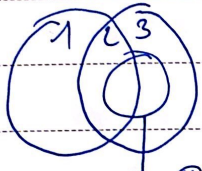


(1): có đặc tả, k° code

(2): _____, có _____

(3): k° _____, _____.

T ⊂ SRS → chỉ test đc tối đa



T ⊂ II.

Ideal

⇒ cần có blackbox và whitebox Testing

Blackbox Testing: + Biên

+ Phân hoạch tương đương

+ Bảng quyết định

Whitebox Testing: + CFT + DFT.

Scanned with CamScanner

$R \subseteq D_1 \times D_2$.

$(x_1, y_1) \in R \mid x_1 \in D_1 ; x_2 \in D_2$.

$\forall x \in D_1 : (x, x) \notin R$.

Đối xứng $(x, y) \in R \Rightarrow (y, x) \in R$.

S M T W T F S

/ /

SUBJECT

⊛ Phân hoạch tương đương :

Q Sinh đc phân hoạch tương đương.

Toán

Lý

Hoá

Biến ra là các giao phân hoạch của từng đặc tả.

3 input → chọn + 1 biến.

Chỉ có tối đa 1 input biến.

Project Management.

⊛ Software Project vs IT Project

Project
- Mục tiêu duy nhất. Hiểu thức, lãnh đạo, khai độ.
- Thực hiện bởi 1 nhóm ng có chuyên môn.
- Tgian bắt đầu + tgian kết thúc (dự kiến)
- Nguồn lực (1x lực, tính phí, ...) (dự kiến)
- Xảy ra trong môi trường bất định

Phân hoạch

TOTO

Scanned with CamScanner

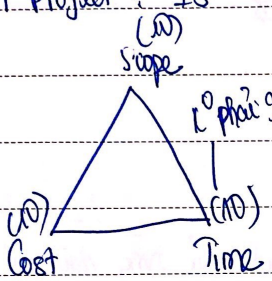
S M T W T F S

/ /

SUBJECT

→ SW Project : Sản phẩm : SW / SW service.

IT Project : IS



↳ phải 9, 14, ... ii có cơ sở tính toán ra con số này.

Tuỳ dự án mà có quy luật.

Dự án có tỷ lệ giữa 3 cái, cùng tăng, cùng giảm

(man/day - m-w-h) bộ 3 constraint (ràng buộc 3 cái kia)

Phải liệt kê từng việc trong dự án → đưa ra các con số theo đó.

Khi thường thường, giảm 1 tiêu chí xuống thì phải giảm cả 2 cái còn lại.

(*) 3 đ nghĩa dự án thành công.

1) Thoả mãn

2) ——— KH - chủ đầu tư

3) ——— các bên liên quan

Khó đáp.

Scanned with CamScanner

S M T W T F S

/ /

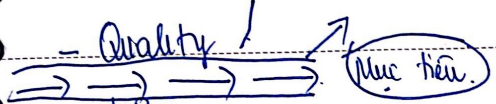
SUBJECT

Trước 2010: có 9 miền tri thức

2010 – nay: – 10 ——— (miền quản lý stakeholders)

- Scope
- Cost
- Time
- Quality

Core



- HR
- Communication
- Risk
- Procurement
- Stakeholders

Project Program

Đảm bảo tích hợp, chia sẻ data, thường đến mục tiêu tổ chức là quan trọng nhất.

TOTO

Scanned with CamScanner