

การกำหนดมาตรฐาน CMMI(Capability Maturity Model Integration)

by TaskS

CMMI (ซีเอ็มเอ็มไอ) เป็นมาตรฐานในการปรับปรุงคุณภาพซอฟต์แวร์ให้มีประสิทธิภาพ เป็นที่รู้จัก และยอมรับของสากล หากองค์กรใดได้รับ CMMI ถือว่าองค์กรนั้นมี product (โปรดัก) และกระบวนการพัฒนา product (โปรดัก) ที่มีประสิทธิภาพ เป็นที่น่าเชื่อถือของลูกค้า และเป็นตัวการันตีชิ้นงานCMMI ให้มีความสำคัญกับ Direct Artifact คือเน้นกระบวนการและผลลัพธ์ของงานมากกว่ารายงานหรือเอกสาร

“Processes allow you to align the way you do business. They allow you to address scalability and provide a way to incorporate knowledge of how to do things better. Processes allow you to leverage your resources and to examine business trends.”

Levels	Continuous Representation Capability Levels	Staged Representation Maturity Levels
Level 0	Incomplete	N/A
Level 1	Performed	Initial
Level 2	Managed	Managed
Level 3	Defined	Defined
Level 4	Quantitatively Managed	Quantitatively Managed
Level 5	Optimizing	Optimizing

ที่มา:

https://www.researchgate.net/profile/Fatih_Yucalar/publication/228676350/figure/tbl1/AS:670522174107664@1536876352373/Capability-and-Maturity-Levels-of-CMMI.png

Standard Operation Procedures (SOPs) คืออะไร ทำไมต้องมี SOPs

SOP เป็นคู่มือหรือเอกสารที่แนะนำวิธีการปฏิบัติงานต่างๆ เพื่อให้องค์กรมีการปฏิบัติงานอย่างถูกต้องและมีทิศทางเดียวกันทั้งหมด โดยระบุขั้นตอนการปฏิบัติงานที่ชัดเจน และสามารถปรับปรุงพัฒนาได้ตามความเหมาะสมของแต่ละแผนกและองค์กร เพื่อให้เกิดผลจริงที่ปฏิบัติได้



ที่มา: <https://checkflow.io/writing-standard-operating-procedures>

Standard Operating Procedures (SOP) Software

1 Free Templates	2 Sharing & permissions
3 Document Control workflow process	4 Centralized processes and procedures
5 SOP Writing & Manuals	6 Dashboard

<https://www.predictiveanalyticstoday.com/top-standard-operating-procedures-sop-software/>

กำหนดหลักการเขียนโปรแกรมและฐานข้อมูลที่มีการพัฒนาใหม่ โดยจัดทำคู่มือมาตรฐาน

ตัวอย่าง การกำหนดมาตรฐาน - หลักการเขียนโปรแกรม

1) หลักการตั้งชื่อ ให้สื่อความหมายรากศัพท์เดิม

1.1 ตัวแปรในโปรแกรมให้กำหนดตัวเล็กทั้งหมด หรือตัวใหญ่ทั้งหมด

ตัวอย่าง ตัวแปร \$V_PAYMENT; or \$V_PM /* variable PayMent */

1.2 การตั้งชื่อ Function or Procedure ให้ใช้ตัวแปร 3 ตัวอักษร ตามการออกเสียงตามคำ ของการถ่ายทอด object หรือ function เช่น

ตัวอย่าง function getdatafirst() or function F_GETDATAFIRST()

```
function gdf() or function F_GDF(); /* function GetDataFirst */
```

การกำหนดคำอธิบาย Remark ในรูปแบบ

ตัวอย่าง

```
/*
```

GDF หมายถึง function GetDataFirst ที่ทำหน้าที่ชี้ข้อมูลไปยังแถวแรก

Input =

Process =

Output =

```
*/
```

2) กำหนด Config มาตรฐานและความยืดหยุ่นในการติดตั้งระบบซอฟต์แวร์

ตัวอย่าง

```
/* Default Language */
```

```
$config['language'] = 'english';
```

3) กำหนด Framework ที่ใช้ ทั้งในส่วน Front-End and Back-End

ตัวอย่าง

3.1 Front End Framework เช่น Bootstrap, Vue.js, React.js, JQuery Angular-JS เป็นต้น

3.2 Back End Framework เช่น Meteor.js, Express.js, YII2, Codeigniter, Laravel, Django, Spring boot เป็นต้น

4) กำหนดรูปแบบการเขียน รวมถึงสรุป Procedure and Function ใช้งาน

5) กำหนด Folder ไม่เกิน 3 ระดับ และตั้งชื่อให้สื่อความหมายรากศัพท์เดิม

6) กำหนดการจัดกลุ่มโปรแกรมย่อยเฉพาะให้อยู่ในกลุ่มเดียวกัน

7) กำหนดวิธีการติดตั้ง การใช้งานของระบบ และจัดทำคู่มือ API และการเรียกใช้งาน

8) กำหนดผู้ทดสอบและเครื่องมือในการทดสอบประสิทธิภาพของระบบงาน

ตัวอย่าง การกำหนดมาตรฐาน - ฐานข้อมูลและตาราง

1) ชนิดฐานข้อมูลตารางต้องเป็น utf8mb4_general_ci (Default)

เหตุผล คือ สามารถเรียกใช้สัญลักษณ์ *Emoji* เป็นมาตรฐาน UTF-8 ขนาด

2) ชื่อตารางใช้ _ เป็นตัวกันคำเช่น user_group

3) กำหนดการแบ่งกลุ่มชื่อตารางออกเป็นสัดส่วน เช่น

ตัวอย่าง - user

- user_info
- user_detail
- master_prefix
- master_province
- master_amphur
- api_token
- api_user

4) การสร้างตารางต้องมี Comment ทั้ง table และ fields

ตัวอย่าง

-- Table structure for table `master_amphur`

```
CREATE TABLE `master_amphur` (
```

```
  `AMPHUR_ID` int(5) NOT NULL, COMMENT 'รหัสลำดับ',
```

```
  `AMPHUR_CODE` varchar(4) NOT NULL COMMENT 'รหัสอำเภอของมหาดไทย',
```

```
  `AMPHUR_NAME` varchar(150) NOT NULL COMMENT 'ชื่ออำเภอ'
```

```
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COMMENT='อำเภอ';
```

5) ตรวจสอบการตั้งค่า NOT NULL หรือไม่ ทุกครั้ง

ตัวอย่าง - `AMPHUR_ID` int(5) NOT NULL, COMMENT 'รหัสลำดับ'

6) ตารางต้องประกอบด้วย 4 fields ทุกตารางดังนี้

ตัวอย่าง - create_date default current_timestamp (NO UPDATE)

- create_by (FK ของ user)

- update_date

- update_by (FK ของ user)

7) ตารางต้องมี rowid เป็น AUTO INCREMENT ทุกตาราง

7.1 INT UNSIGNED, it ranges is 0 to 4294967295.

7.2 BIGINT UNSIGNED, it range is 0 to 18446744073709551615.

8) พยายามทำ INDEX หรือ UNIQUE ทุกครั้งโดยพิจารณาดังนี้

ตัวอย่าง - field FK ไหนที่สามารถมีค่าซ้ำได้ให้ใช้ INDEX

- field FK ไหนที่ไม่สามารถมีค่าซ้ำได้ UNIQUE

- ทำ INDEX ตารางและ Field ควรมี create_date, create_by, update_date, update_by ทุกตาราง

9) การตั้งชื่อ Views ต้องเป็น v_ ขึ้นต้น

ตัวอย่าง - CREATE VIEW v_business_type

10) การตั้งชื่อ Function ต้องเป็น F_ ขึ้นต้น

ตัวอย่าง - ตัวแปรของต้องเป็นพิมพ์ใหญ่ทั้งหมด

- ตัวแปรของ Parameters ต้องขึ้นต้นด้วย P_

- ตัวแปรปกติใช้ V_ ขึ้นต้น

10) การตั้งชื่อของ Procedure ต้องขึ้นต้นด้วย SP_

11) การตั้งชื่อ Event Schedule ต้องขึ้นต้นด้วย E_

12) จัดทำคู่มือ API และการเรียกใช้งานตารางข้อมูลการรับส่งข้อมูล

ตัวอย่าง คู่มือส่วนต่อประสานโปรแกรมประยุกต์ (Application program interface: API)

ระบบการบริหารทรัพยากรมนุษย์ (Human Resource Management: HRM)

Web service - การรับส่งข้อมูล ระหว่าง ผู้ใช้กับการบริหารทรัพยากรมนุษย์

Service type - REST API [JSON]

Version – 1.0

1) เข้าใช้งานระบบ

URL: <http://hrm.service.com/api/v1/authorize>

METHOD: POST

Input Data (Type JSON)

No. Field Name Field Type Field Size Required Description

1.	username	Text	20	Y	ชื่อผู้ใช้งาน
2.	password	Text	20	Y	รหัสผ่าน

Output Data (Type JSON)

No. Field Name Field Type Field Size Required Description

1.	token	Text		Y	รหัสข้อมูลสำหรับเข้าสู่ระบบ
----	-------	------	--	---	-----------------------------

2) ตรวจสอบสถานะ

URL: <http://hrm.service.com/api/v1/statuscheck>

METHOD: GET

Output Data (Type JSON)

No. Field Name Field Type Field Size Required Description

1	message	Text			ข้อความรายละเอียด
2	status	Text			ผลการเรียก API (true/false)

การกำหนดมาตรฐาน – UX/UI

คำชี้แจง				
ห้าม ใช้งานปุ่มใหญ่เกิน class btn-lg (หรือขนาด 147x48px) หากใหญ่กว่านี้ต้องออกแบบ UI ใหม่ห้ามนำ button มาใช้งาน				
ลำดับ	คำสั่ง ความหมาย	Color		note
		Class	Code	
1	button ทั่วไปที่เป็นองค์ประกอบของเว็บ (ไม่มี ความหมายตายตัว) เช่น กดเพื่อ Process กระบวนการต่าง ๆ แบบ Dynamic function หรือการ Download ข้อมูลต่าง ๆ (ที่ไม่ใช่ Image, PDF, Word, Excel) เป็นต้น	btn-primary	#007bff	Blue
2	Insert ข้อมูล Save งาน Successfully หรืออื่น ๆ ที่เกี่ยวกับการทำงานสำเร็จ	btn-success	#28a745	Green
3	เปิด Modal (ป๊อปอัพ) Draw หน้า HTML Check ข้อมูล Search data รวมทั้งหมด แสดงผลรายการข้อมูล หรือแสดงรายละเอียดต่าง ๆ *หมายเหตุ หากหน้านั้นเป็นข้อมูลเกี่ยวกับ History, Trash, Temp, Log ให้ไปใช้กฎข้อ 4	btn-info	#17a2b8	Blue sky
4	Danger (ความเสี่ยง อันตราย ไม่ปลอดภัย) Delete ข้อมูล การ Remove element หรือ attribute ต่าง ๆ ความเร่งด่วน ความสำคัญที่ต้องคำนึง Cancel ยกเลิก interrupt ชัดแจ้งหยุดการทำงานของระบบ และอื่น ๆ ที่ผู้ใช้งานจะต้องให้สำคัญในการ ตัดสินใจให้รอบครอบก่อนทำการคลิก	btn-danger	#dc3545	Red
5	ไว้เรียกหรือแสดงผลข้อมูลที่เกี่ยวข้องกับ History, Trash, Temp, Log ข้อมูลการ ref. หรือข้อมูลชั่วคราวต่าง ๆ *ประมวลผลข้อมูลที่เป็น อดีต	btn-warning	#ffc107	Orange
6	ใช้กับงานที่เกี่ยวข้องกับการแสดงผล หรือดาวน์โหลด Image, PDF, Word, Excel การแสดงผลรายงาน ออกรายงาน Output	btn-secondary	#6c757d	Gray