

自動車開発における ソフトウェアテスト完了判断と 市場故障発生予測のモデル に対する検証

～オーカマモデル～

2021-02-26

ソフトウェア品質管理研究会SQiP

研究コース1 ソフトウェアプロセス評価・改善 分科会

研究員：大釜俊洋（本田技研工業株式会社）

主査：山田淳（株式会社東芝）

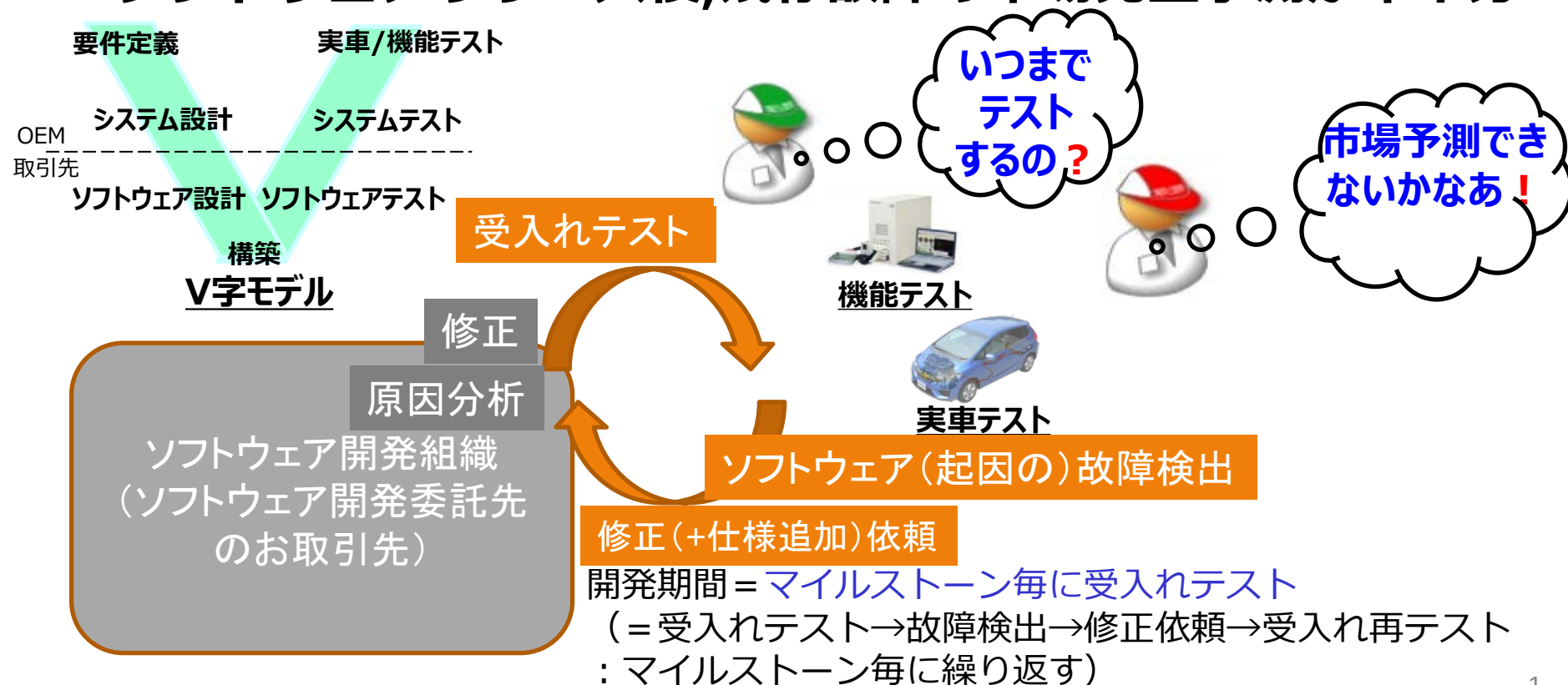
副主査：田中桂三（オムロン株式会社）

アドバイザー：中森博晃（パナソニック株式会社）

1. 研究の背景（現状）と問題

自動車開発における、ソフトウェア開発委託取引先から受け入れてテストを実施する際に問題がある：

- ・ ソフトウェア故障の残存発生数の予想が困難
- ・ テスト完了の判断方法・完了判断の定量的な基準が不明確
- ・ ソフトウェアリリース後、残存故障の市場発生予測が不十分

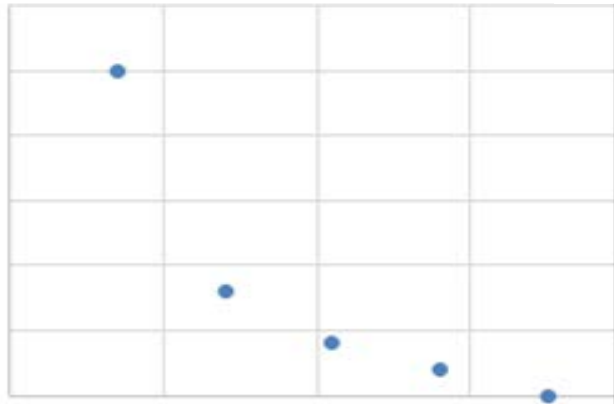


テストでの故障発生傾向

収束する

目標テスト
期間

故障
数



開発期間(マイルストーン毎に受入れテスト)

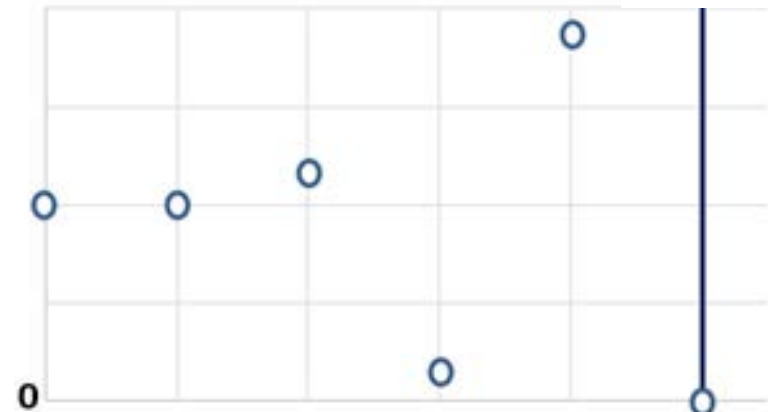
図1-1

①検出した故障の件数が収束して信頼度成長曲線も適用しやすい場合

収束しない!?

目標テスト
期間

故障
数



開発期間(マイルストーン毎に受入れテスト)

図1-2

②残存故障の検出が困難で故障件数がなかなか収束しない場合

2. 問題解決のアプローチ

1. **テスト完了**を数値で定量的に判断するために,受入れテスト中に算出される**故障の発生率** (k : 故障率) に着目し,**ソフトウェア信頼度成長モデル**と組合せることで,市場での故障発生を予測し**テスト完了の判断基準**を明確にする.
2. テスト工程中の**故障が収束していない場合**,収束するための**原因分析**.

3. 提案するオーカマモデルの全体像

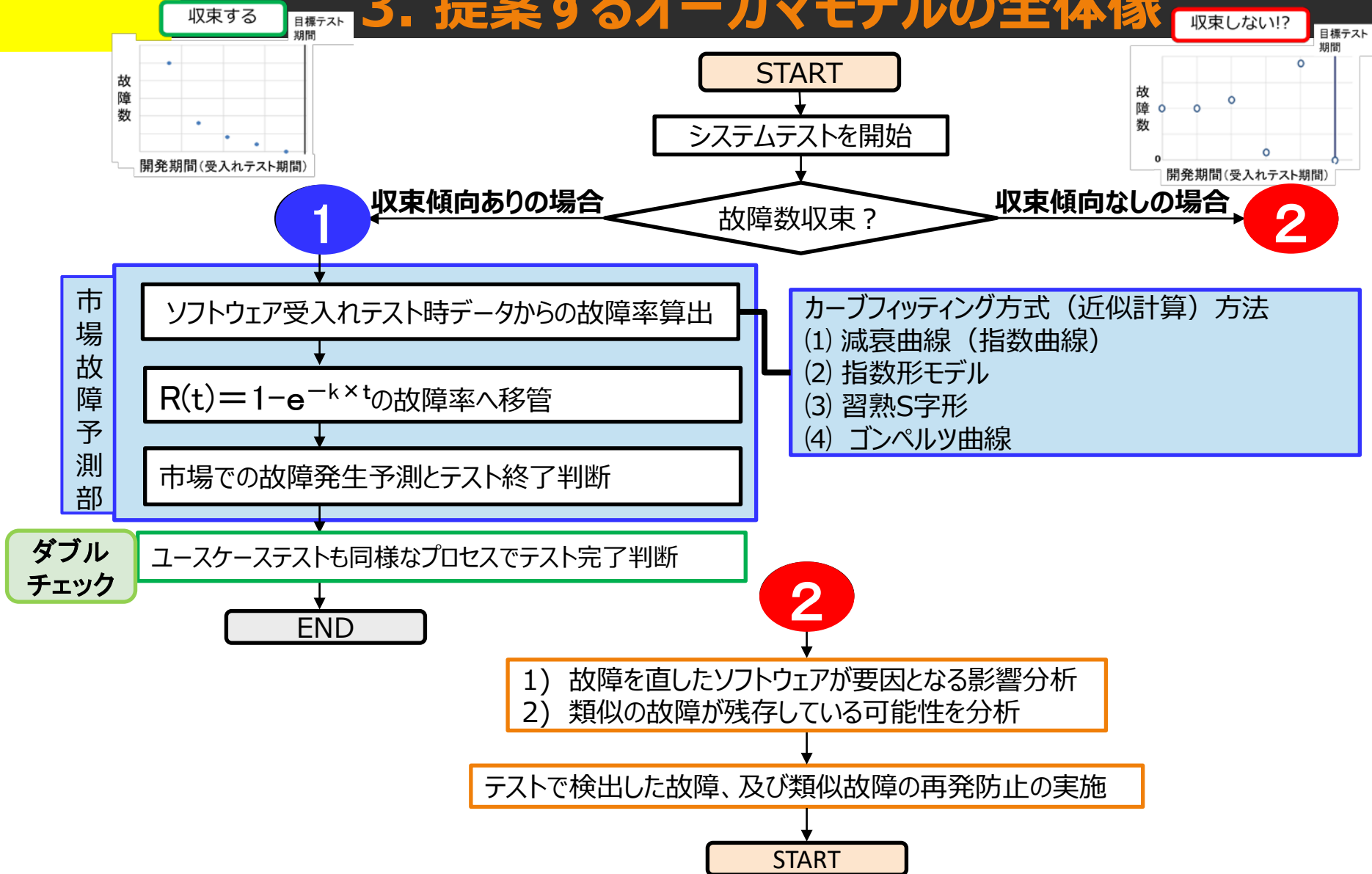


図3-1 ソフトウェアテスト完了判断と市場故障発生予測のモデルで用いる方法（オーカマモデル全体）の手順

3.1テスト工程の故障収束判断と故障数の経時変化をモニター（検証事例）



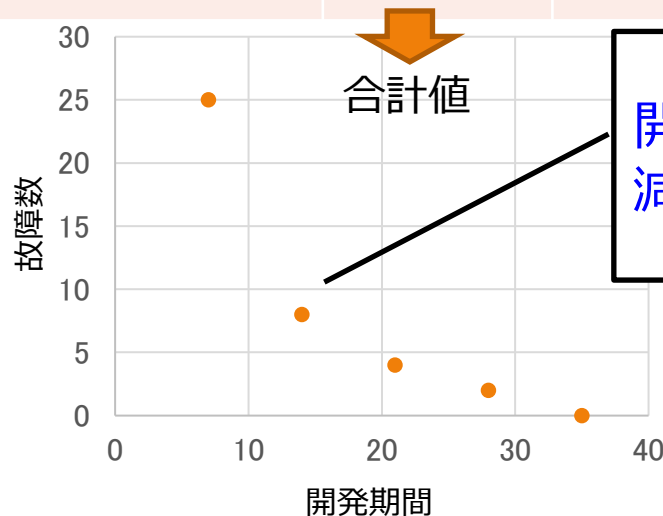
◆収束傾向？

開発中の典型的な値を検証に用いた。

開発期間	各マイルストーン期間	マイルストーン回数	テストケース	レビュー項目数	市場保証期間	市場生産台数	保証件数
35か月	7か月	5回	100	30	10年	100万台	1件未満

（要求仕様の追加仕様もマイルストーン毎には減少する）

	マイルストーン	1回目 7月	2回目 14月	3回目 21月	4回目 28月	5回目 35月
動的テスト	テスト時故障数	10	3	2	1	0
静的テスト	レビュー指摘数	15	5	2	1	0



開発マイルストーン毎の故障数は、減衰曲線を描き、収束傾向にある

1

図3-2 開発マイルストーン毎の故障数

3.2 減衰曲線（指数関数）とソフトウェア信頼度成長モデルで故障累積数をカーブフィッティングした推定曲線から故障率を算出

◆故障率kを計算

1

1. 実故障と指数減衰曲線推定（最小二乗法で補正前処理）の差分の二乗和を最小化
2. 最小二乗法で補正前処理したカーブの傾き（故障率(t)）の時間変化

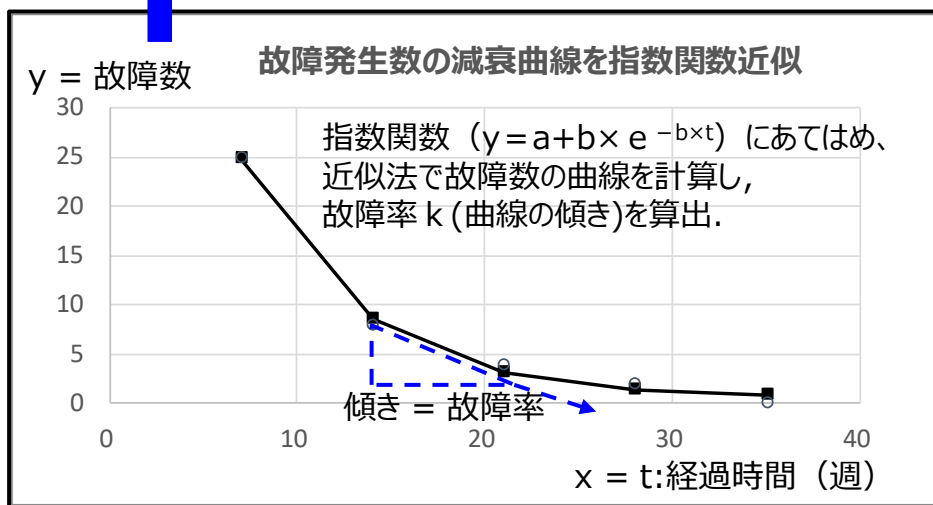
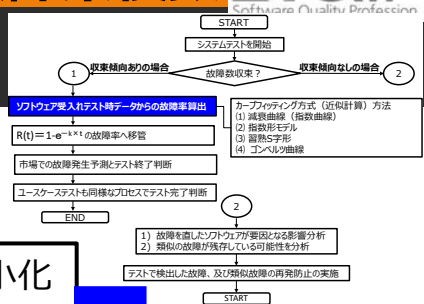


図3-3a 故障発生数の減衰傾向を指数関数の減衰曲線で近似

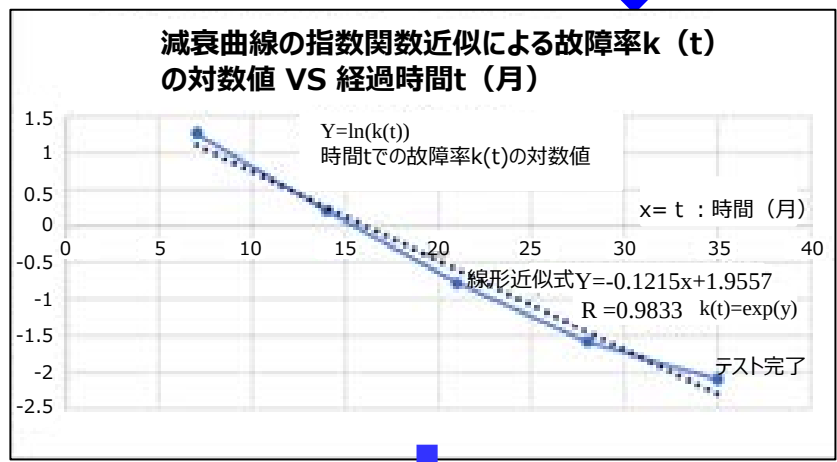


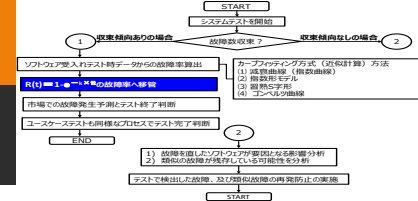
図3-3b 対数変換して線形近似

各ソフトウェア信頼度成長モデルから故障率とMTBF算出

(1) 減衰曲線（指数曲線）	$F(t) = a + b \times e^{-b \times t}$	故障率（件/月）	MTBF(月)
(2),(3),(4)の故障率k(t(i))は時間t(i-1（35か月）)からt(i（36か月）)の間の単位時間の傾き			
(2) 指数形モデル	$F(t) = a \times (1 - e^{-b \times t})$	0.0390	25.7
(3) 習熟S字形	$F(t) = a \times (1 - e^{-b \times t}) / (1 + c \times e^{-b \times t})$	0.0390	25.7
(4) ゴンペルツ曲線	$F(t) = a \times \exp(-b \times e^{-c \times t})$	0.00711	140.7

1/故障率

3.3 $R(t) = 1 - e^{-k \times t}$ の故障率へ移管



開発期間

市場

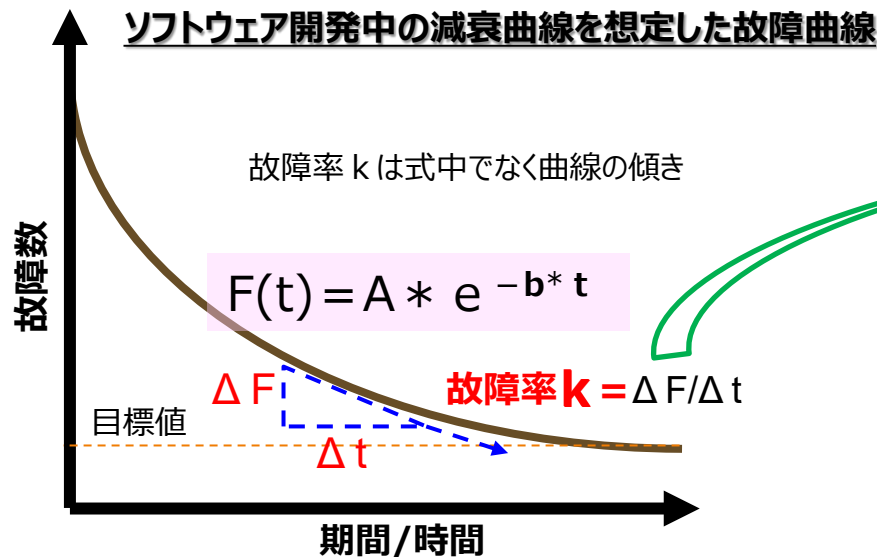
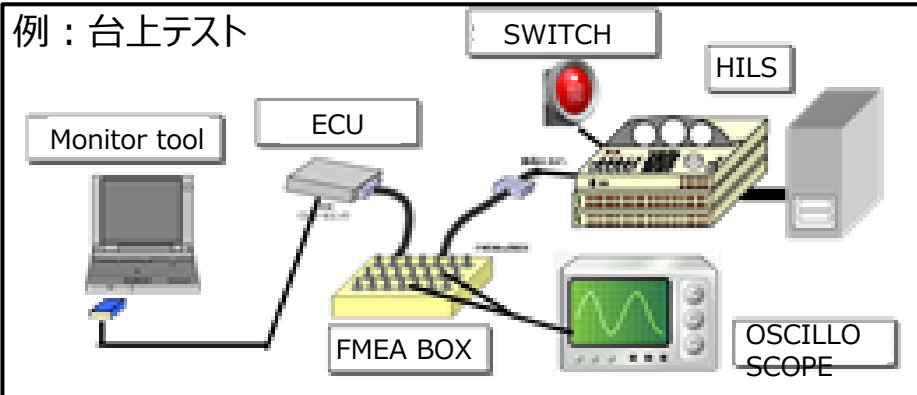


図3-4a ソフトウェア開発期間（開発期間）

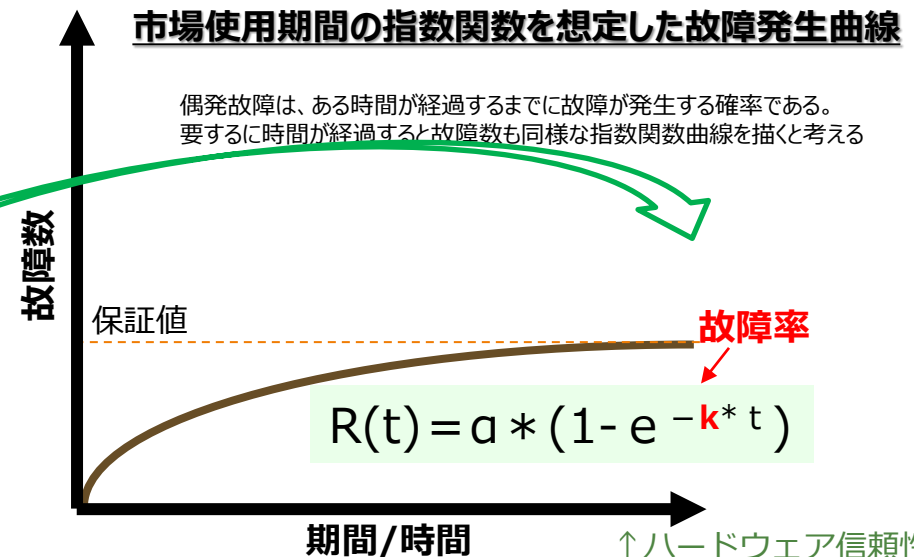
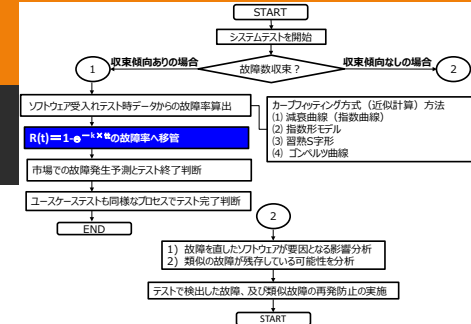


図3-4b 市場使用期間

市場での故障発生を予測しテスト完了の判断基準を明確にする。

3.4 市場での故障発生予測の計算



◆市場での故障発生予測

算出した故障率を用いて、 $R(t)=a \times (1 - e^{-k \times t})$ の故障率 k に置き換え、市場での故障発生予測をする。

表3-1 算出法別の故障率 k 、MTBF と市場故障数予測一覧

$$R(t) = a \times (1 - e^{-k \times t})$$

	故障率 k (件/月)	MTBF (月)	その時点での MTBF > テスト完了後の市場保証期間10年完了時 (35+10 x 12=155月) となるまでの、テスト開始からの経過時間 (年)	100万台中10年間の市場故障予測 (台)
(1) 減衰曲線 (指数曲線)	0.089	11.2	4.6	999,977
(2) 指数形モデル	0.0390	25.7	3.9	990,710
(3) 習熟S字形	0.0390	25.7	3.9	990,710
(4) ゴンペルツ曲線	0.00711	140.7	3.0	573,951

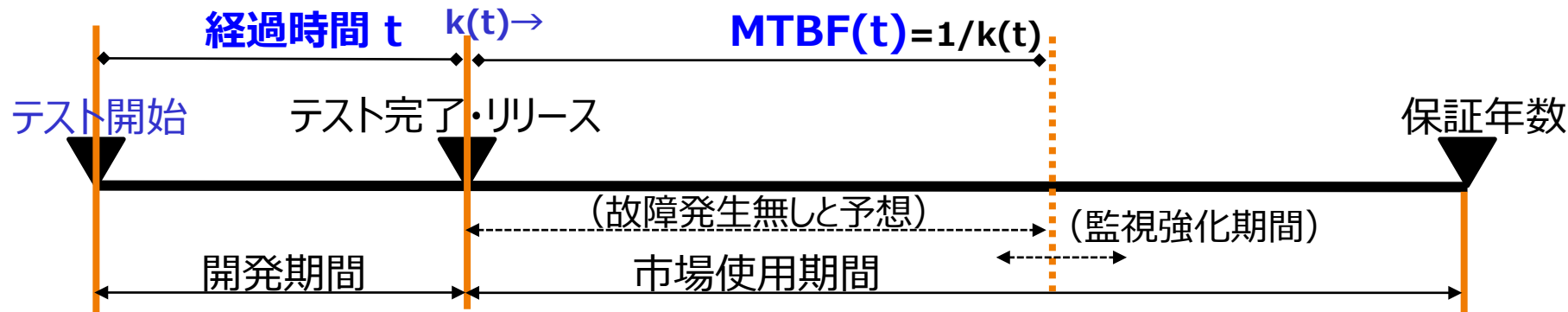


図3-5 テスト完了直後の $k(t)$ 、 $MTBF(t)$ と保証年数

```

graph TD
    START([START]) --> ShiftCost[シフトコスト計算]
    ShiftCost --> Branch1{ }
    Branch1 --> Case1((1))
    Branch1 --> Case2((2))
    
    Case1 --> CalcOpt1[ソフトウェアを用いた2ステップからの最適値算出]
    CalcOpt1 --> Formula[式(1) =  $1 - e^{-\frac{1}{2} \times \frac{1}{\text{シフトコスト}}}$  の値を代入]
    Formula --> Table1[表2での最適値を予測値として算出]
    Table1 --> Check1{ }
    Check1 --> End([END])
    Check1 --> Case2
    
    Case2 --> CalcOpt2[トープシフトメソッド(近似計算)方法]
    CalcOpt2 --> List1[① 構成単位 (構成数群)]
    List1 --> List2[② 各単位内数 (構成数群内数)]
    List2 --> List3[③ 変数の数]
    List3 --> Table2[表3のシフトコスト]
    Table2 --> Check2{ }
    Check2 --> Case1
    Check2 --> Note[テストで検証した結果、より適切な収束経路の再決定が必要]
    Note --> START
  
```

故障数累積値 (件)



ソフトウェア故障は本来台数とは無関係だが、十分にテストされたソフトウェアでも、市場で長時間使用される環境下では、テストケースで網羅されていない条件が揃う可能性が増しソフトウェア故障が確率的に発生する場合がある。このため $R(t) = a \times (1 - e^{-k \times t})$ による市場での故障累積数予測と設定保証値との比較は、リスクを表現する手段の一つとして考えるのがよい。

1) 故障率の算出

指数減衰曲線推定
(最小二乗法で補
正前処理)

最小二乗法で補正
前処理したカーブの
傾き (故障率(t))
の時間変化

減衰曲線の指数関
数近似を適用
 $y = \ln(k(t))$



故障率をよりの確に
算出できる可能性あり

図3-7 故障率の算出 (指数減衰曲線推定の場合)

⇒ 複数の算出法 (指数減衰曲線+信頼度成長モデル) を組合せて一覧し、
比較又は選択して用いるのが良いと考える。

2) 市場故障予測の保証値(=1件)未満には達しなかった。

⇒ 故障率のGapから定量的な市場リスク指標 ⇒ テスト完了判断材料

⇒ 故障台数累積予測式 $R(t) = a \times (1 - e^{-k \times t})$ によるソフトウェアの市場故障数予測値は、市場リスクを表現する手段の一つとして考えるのがよい

さらに下記の対策が考えられる。

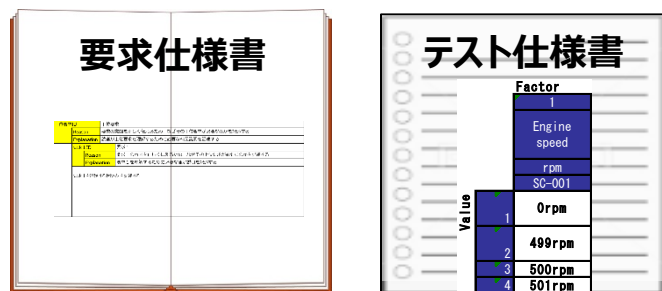
(a)保証値を達成する故障率になるまで継続してテストを実施し、検出故障を確実に対策する

(b)テスト終了後に**MTBF**の期間毎に市場監視し、テスト長期化を避ける

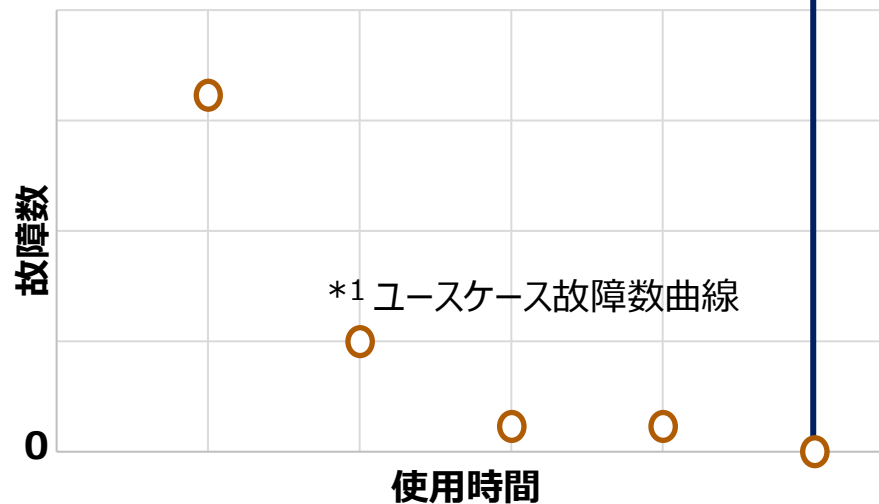
(c)ソフトウェア開発組織内のデータを蓄積、活用することで推定精度を上げる

[illegible]

ダブル チェック



目標テスト時間



11

ユースケーステストにおける故障数を図3-9に示す。
*1ユースケース故障数曲線から減衰曲線の指数関数近似の適用の可能性を示唆しており、
 $R(t) = a * (1 - e^{-k * t})$ を用いることで、
市場故障発生数予測の可能性に期待できる。

3.7 故障数が収束していないケース



◆収束傾向なしの場合

2

テストを実施し故障数が収束していないケースを図3-10に示す
 ソフトウェアに欠陥が残存している可能性が高い
 →ソフトウェア故障発生の可能性も高い

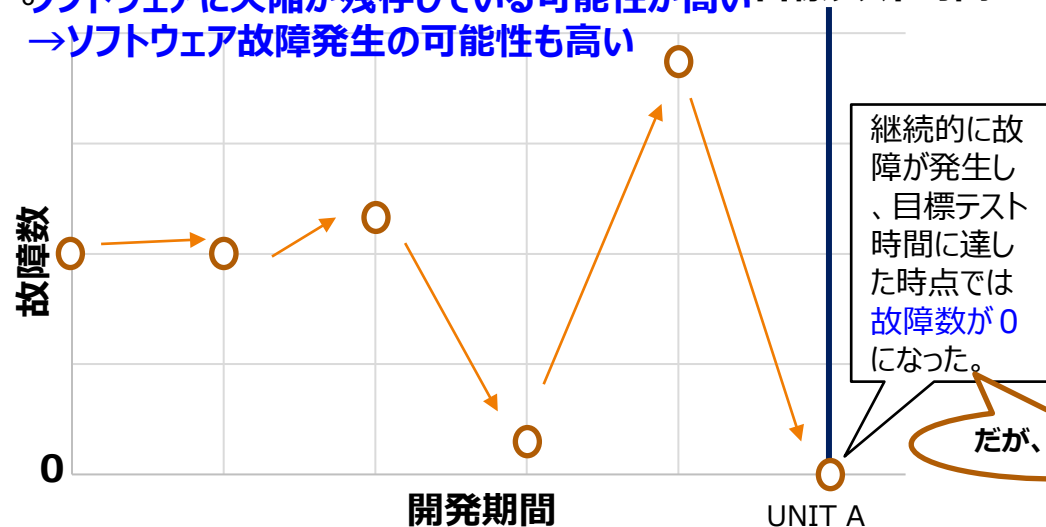


図3-10 UNIT A 故障数曲線

UNIT A
 ROM使用量：796K
 RAM使用量：35K

改善策

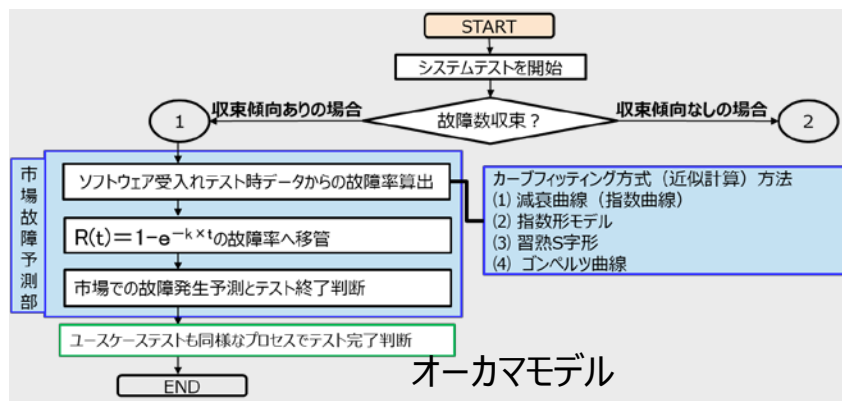
- 1.故障の原因を分析し再発防止を実施
- 2.故障を直したソフトウェアの影響分析を実施

児島だよチーム
 “ARC
 メソッド”

あつ森チーム
 “あつ森法”

原因コード

要求仕様書の確認不足
要求仕様書の理解不足
関連機能の確認不足
関連機能の理解不足
テスト仕様書の確認不足
テスト仕様書の理解不足
テスト方法習熟不足
テスト仕様書の検討不足



オーカマモデル

改善策を講じ、システムテストを開始し、故障数収束を確認する

4. 結論

ソフトウェア信頼度成長モデルから故障率を算出し, $R(t) = 1 - e^{-k \times t}$ に置換する**オーカマモデル**を用いることで,市場故障発生予測と関連付けて, **より精度良くテスト完了・追加の判断へとつながることが期待できる。**

また,**MTBF**より算出した平均故障時間より, **定期的な市場監視を推奨できる。**



図4-1 テスト+定期的市場監視

5. 今後の課題と展開～おわりに

- 1) **故障率の精度向上**,市場故障発生**予測の的確性**を求めるために,ソフトウェア信頼度成長モデルの**更なる追究**.
- 2) **原因分析と対策**及びテストケース内容との**関連性**,またソフトウェア**故障区分別の潜在故障予想**及びテスト完了判断指標の策定.

ご清聴ありがとうございました。



APPENDIX

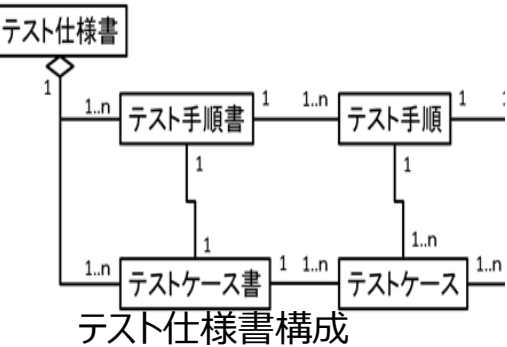
表2 テスト故障原因分析

原因コード	原因コードの推定材料
要求仕様書の確認不足	① Input情報の入手 1. 確認したInput情報バージョン 2. 確認したInput情報の網羅性 3. 確認したInput情報の伝達 ② 要求仕様書の一致性
要求仕様書の理解不足	① 要求仕様書の合意
関連機能の確認不足	① 利害関係者の期待事項 ② システム要求、ソフトウェア要求の構造化 ③ システム要求、ソフトウェア要求の影響 ④ システム要求、ソフトウェア要求の構造化の一致性
関連機能の理解不足	① システム要求、ソフトウェア要求の構造化の合意
テスト仕様書の確認不足	① テスト戦略 ② テスト設計書 1. テスト仕様書 2. テスト手順書 3. テストケース
テスト仕様書の理解不足	① テスト設計書の合意
テスト方法習熟不足	① テスト担当者がテスト習熟しなかったことに気付いたテスト手順書があるケース
テスト仕様書の検討不足	① 上記に該当しなかった場合は、『テスト仕様書の検討不足』が原因コード

テスト仕様書の概要

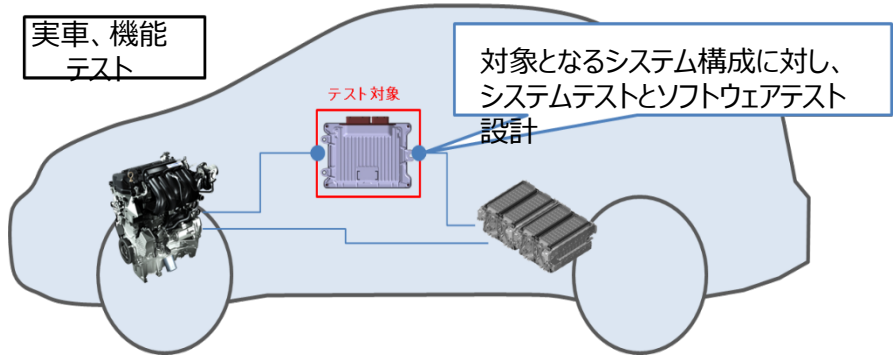
テストケース書は、要求仕様書からテスト対象の機能、シナリオ、期待値（判断基準）、因子、水準を抽出し作成される

テスト仕様書は、「テスト手順書」と「テストケース書」で構成



No	確認内容(シナリオの内容)	期待値(判断基準)	因子	水準	1	2	3	4	5	6	7	8	9	10	11	12	
1	Start Stop switchを押してすぐ離れた(長押ししていない)場合のSTS信号は入力される。	STS信号は、0v-バッテリー電圧、730ms±20ms以外のこと	電圧	0v													
5.9v(SPEC外)				○	○	○	○										
6v								○	○	○	○						
12v(SPEC外)												○	○	○	○		
ON時間			709m(SPEC外)	○				○					○				
			710ms		○				○					○			
			750ms			○				○					○		
			751ms(SPEC外)				○					○					○
テスト区分																	
Autoモード																	

テストケース書例



対象機能環境下におけるテストとテスト対象システム例

従来の代表的SRGMとそのY方程式

※SRGM: software reliability growth model

モデル (曲線)	一般式	1), 2) 統合モデルの微分方程式のパラメータ				一次微分 y'	Y方程式
		α	β	γ	δ		
指数形	$y = N(1 - e^{-bt})$	Nb	b	1	0	$Nb e^{-bt}$	$Y_3 + bY_1 - \ln(Nb) = 0$
ゴンペルツ	$y = N \exp(-k e^{-bt})$ (注1)	kb	b	0	0	$kb e^{-bt} y$	$Y_3 - Y_2 + bY_1 - \ln(kb) = 0$
ロジスティック	$y = \frac{N}{1 + \psi e^{-bt}}$	$\frac{b\psi}{N}$	b	-1	0	$\frac{b\psi e^{-bt}}{N} \cdot y^2$	$Y_3 - 2Y_2 + bY_1 - \ln\left(\frac{b\psi}{N}\right) = 0$
習熟S字形	$y = N \frac{1 - e^{-bt}}{1 + \psi e^{-bt}}$ (注2)	$\frac{b\psi^2}{N(\psi + 1)}$	b	-1	$\frac{N}{\psi}$	$\frac{Nb(\psi + 1)e^{-bt}}{(1 + \psi e^{-bt})^2}$	—
遅延S字形	$y = N \{1 - (1 + bt)e^{-bt}\}$	$\approx 2.2 N^{0.46} b$	$\approx 0.78b$	≈ 0.46	0	$Nb^2 t e^{-bt}$	$Y_3 - 0.54Y_2 + 0.78bY_1 - \ln(0.78Nb) = 0$ (近似式)
超指数形	$y = \sum_{i=1}^n N_i (1 - e^{-b_i t})$	$\approx \frac{N\beta}{\gamma}$	—	> 1	0	$\sum_{i=1}^n N_i b_i e^{-b_i t}$	(Y2の係数が正となるY方程式。 ただしY3=1とする)
ワイブル過程	$y = \lambda t^n$	$n\lambda^{\frac{1}{n}}$	0	$\frac{1}{n}$	0	$\lambda n \left(\frac{y}{\lambda}\right)^{\frac{n-1}{n}}$	$Y_3 + \frac{1-n}{n}Y_2 - \ln(n) - \frac{1}{n}\ln(\lambda) = 0$

古山恒夫 ソフトウェア信頼度成長モデルに関する統合モデルの解析的パラメータ推定法 情報処理学会論文誌, 1996

3.2 補足 ソフトウェア信頼度成長モデル故障で累積数曲線をカーブフィッティングした推定曲線から故障率を算出

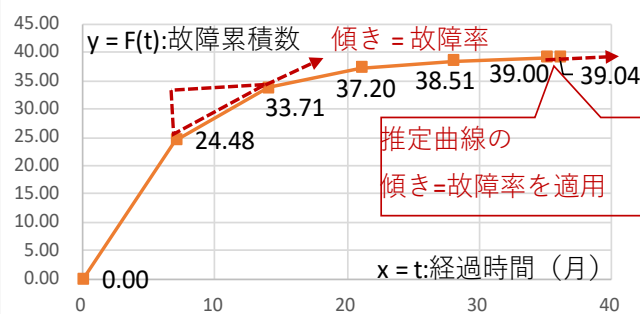
指数形モデル

習熟S字形モデル

ゴンペルツ曲線

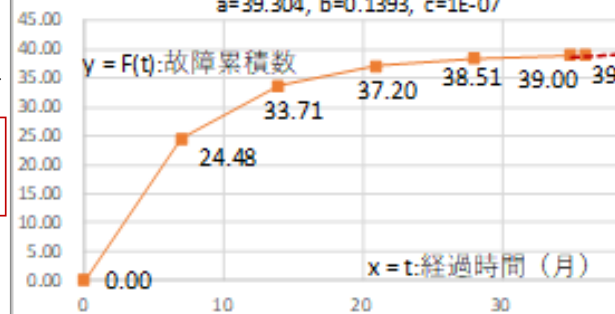
指数形モデルでフィッティング推定

$$F(t) = a \times (1 - \exp(-b \times t)) \quad a=39.304, b=0.1393$$



習熟s字形モデルでフィッティング推定

$$F(t) = a \times (1 - \exp(-b \times t)) / (1 + c \times \exp(-b \times t)) \quad a=39.304, b=0.1393, c=1E-07$$



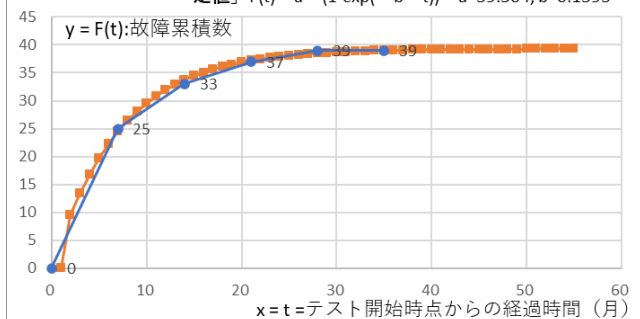
ゴンペルツ曲線でフィッティング推定

$$F(t) = a \times \exp(-b \times e^{-c \times t}) \quad a=40.0000, b=2.85, c=0.2316$$

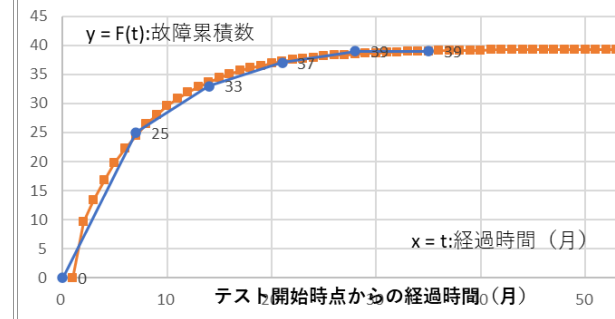


テスト後も含めて推定

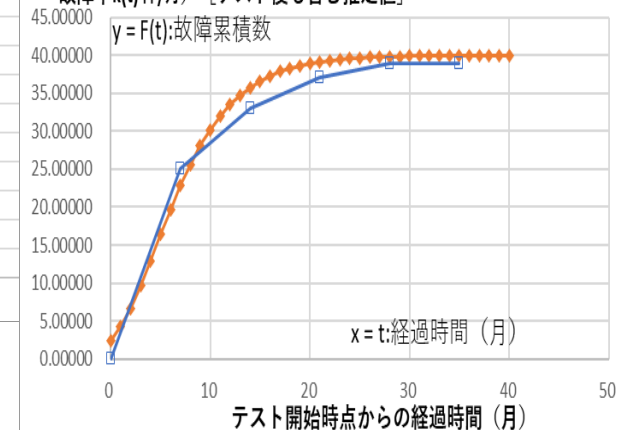
指数形モデルフィッティング推定曲線から計算した故障累積数（傾きが故障率k(t)件/月）[テスト後も含む推定値] $F(t) = a \times (1 - \exp(-b \times t)) \quad a=39.304, b=0.1393$



習熟s字形モデルフィッティング推定曲線から計算した故障累積数（傾きが故障率k(t)件/月）[テスト後も含む推定値]



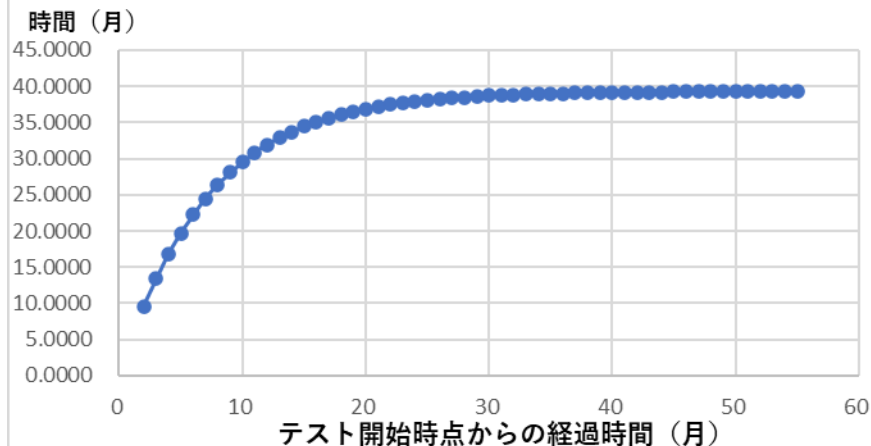
ゴンペルツ曲線モデルフィッティングから計算した故障累積数（傾きが故障率k(t)件/月）[テスト後も含む推定値]



信頼度成長モデルから故障率算出を通して推定MTBFを算出

信頼度成長モデルでフィッティング

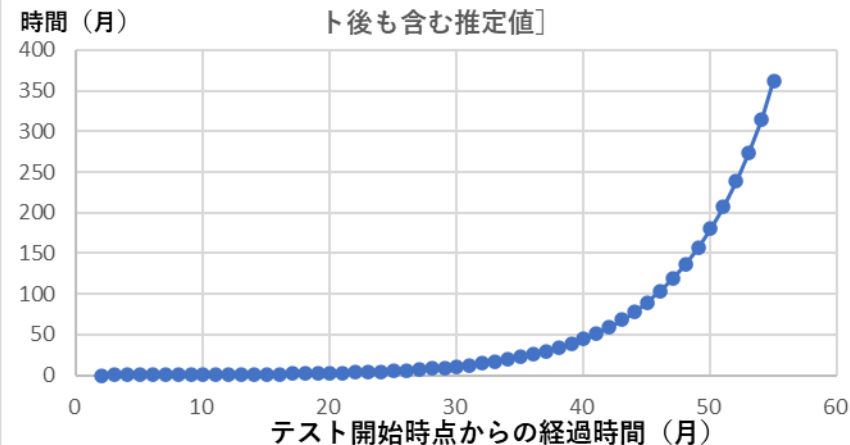
指数形モデルフィッティング推定曲線 [テスト後も含む推定値] $F(t) = a \times (1 - \exp(-b \times t))$ $a=39.304, b=0.1393$



単位時間当たりの傾きから故障率を算出

単位時間当たりの傾きから故障率の逆数から経過時間tでのMTBF(t)を計算

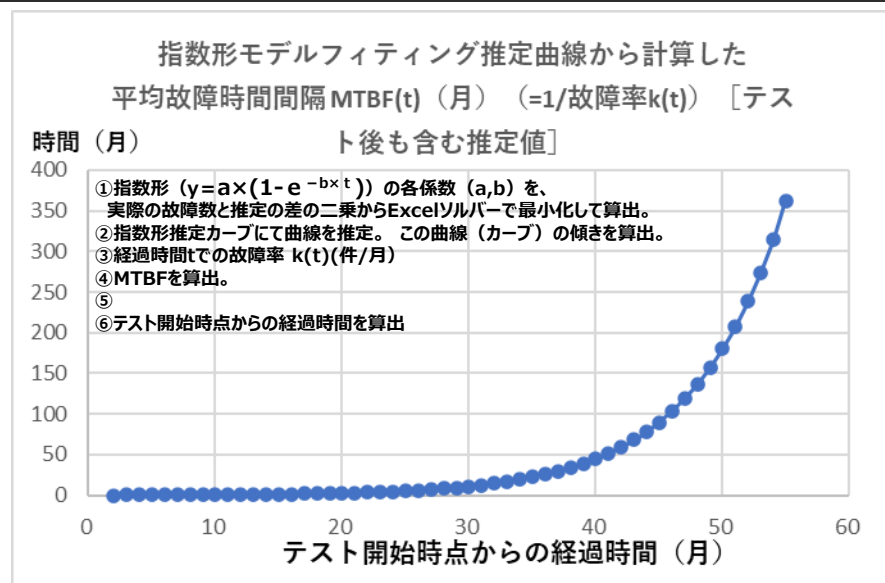
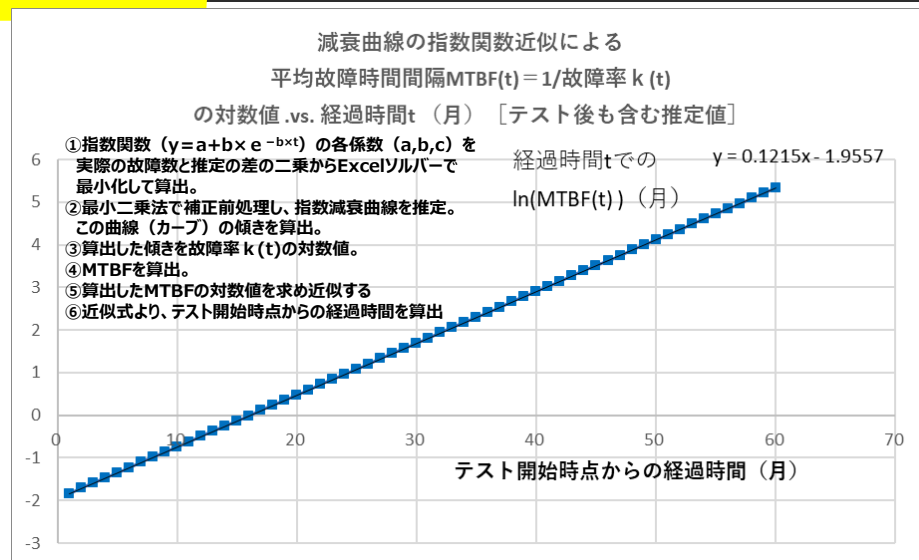
指数形モデルフィッティング推定曲線から計算した平均故障時間間隔 MTBF(t) (月) ($=1/\text{故障率}k(t)$) [テスト後も含む推定値]



$$MTBF = \text{開発時間} / \text{故障回数} \quad (\text{故障率} = 1 / MTBF)$$

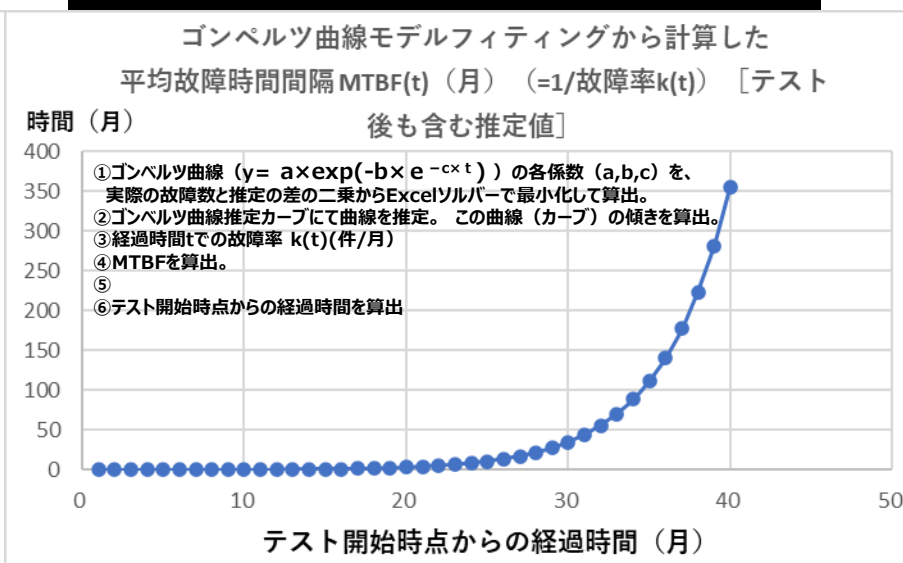
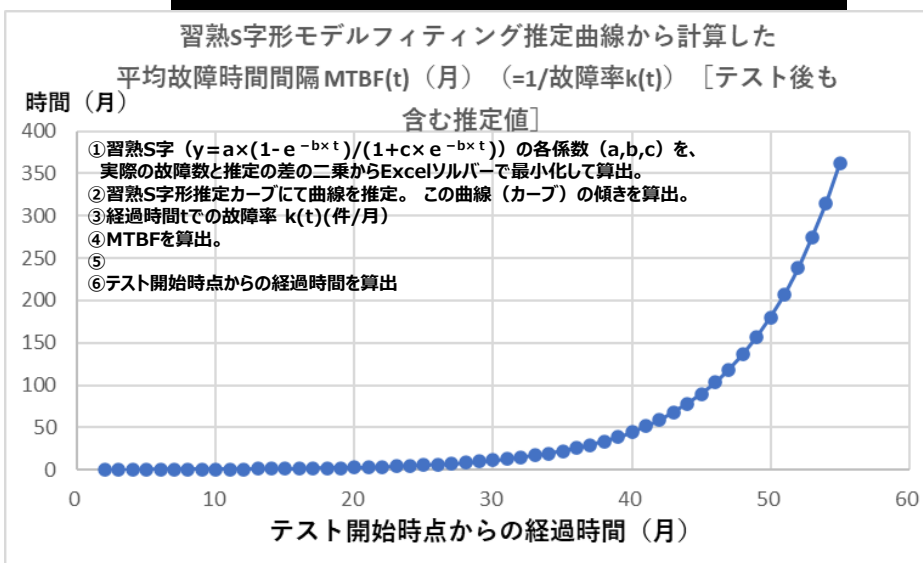
(1)減衰曲線の指数関数近似

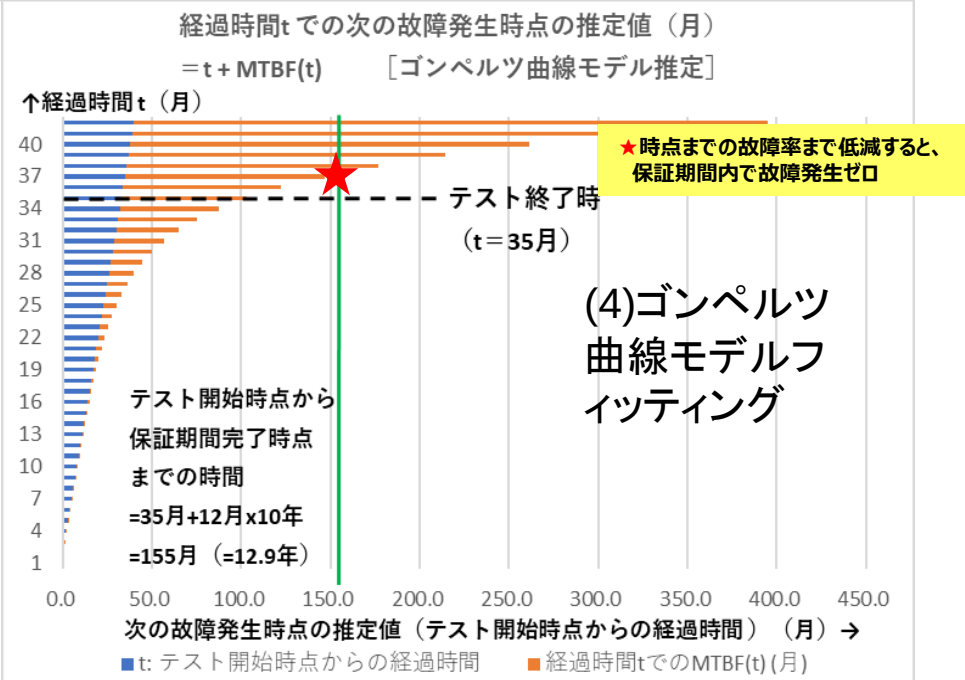
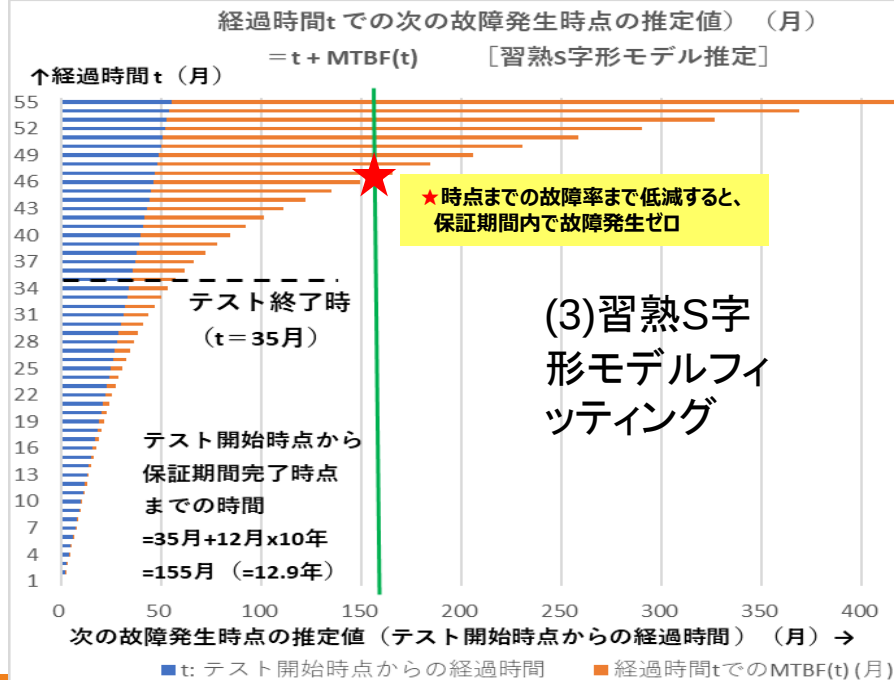
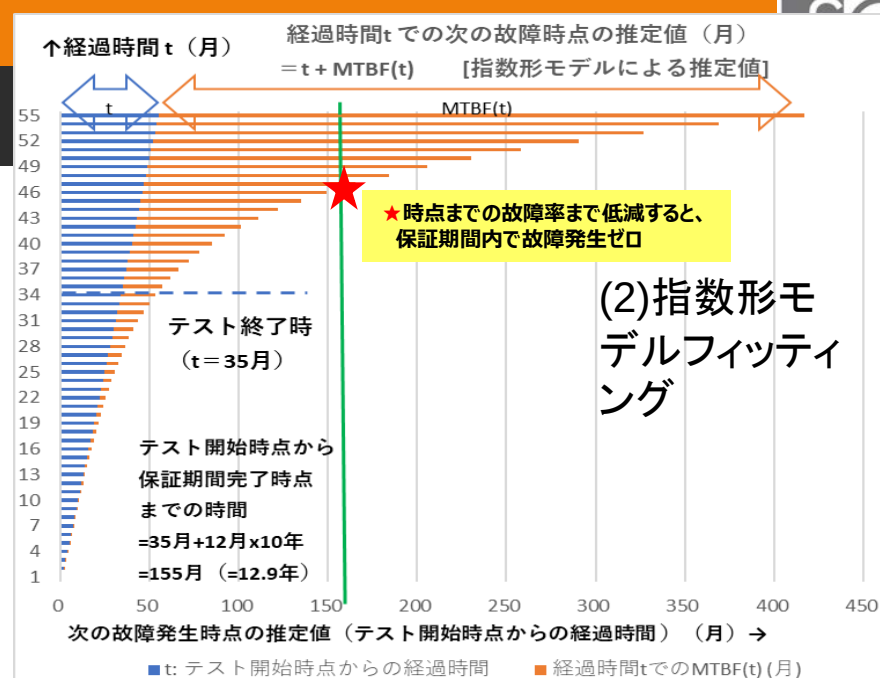
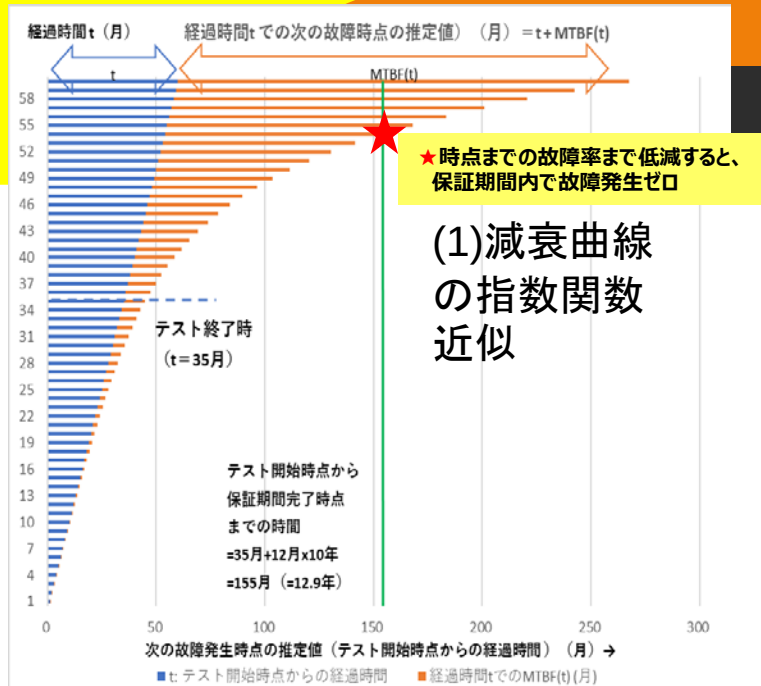
(2)指数形モデルフィッティング



(3)習熟S字形モデルフィッティング

(4)ゴンペルツ曲線モデルフィッティング





1) 故障率の算出解説

指数減衰曲線推定（最小二乗法で補正前処理）

最小二乗法で補正前処理したカーブの傾き（故障率 $k(t)$ ）の時間変化

減衰曲線の指数関数近似を適用 $y = \ln(k(t))$



故障率をより的確に算出できる可能性あり

- ① 指数関数 ($y = a + b \times e^{-b \times t}$) の各係数 (a, b, c) を実際の故障数と推定の差の二乗からExcelソルバーで最小化して算出。
- ② 最小二乗法で補正前処理し、指数減衰曲線を推定。この曲線（カーブ）の傾きを算出。
- ③ 算出した傾きを故障率 $k(t)$ の対数値。
- ④ MTBFを算出。
- ⑤ 算出したMTBFの対数値を求め近似する
- ⑥ 近似式より、テスト開始時点からの経過時間を算出



		①		①	②	③	④	⑤
減衰曲線の指数関数近似による故障率 $k(t)$		$y = A + B \cdot \exp(-C \cdot t)$						
開発期間	35か月	A=	0.579655789					
マイルストーン間隔	7か月	B=	73.98748062					
		C=	0.158909309		最小二乗法で補正前処理したカーブの傾き（故障率 $k(t)$ ）の時間変化	傾き=故障率 $k(t)$ の対数値		
t(月)		故障数	指数減衰曲線推定（最小二乗法で補正前処理）	実際と推定の差の二乗	経過時間tにおける故障率 $k(t)$ / 月	$\ln(k(t))$	$MTBF = 1/k(t)$	
0						#NUM!		
7	25		24.90529089	0.008969816	3.5578986985	1.269170117	0.2810647758	-1.2691701174
14	8		8.577448363	0.333446612	1.2253497662	0.203226327	0.8160935168	-0.2032263266
21	4		3.209173407	0.6254067	0.4584533438	-0.779896751	2.1812470417	0.7798967507
28	2		1.444189701	0.308925088	0.2063128145	-1.578361745	4.8470086679	1.5783617452
35	0		0.863897614	0.746319087	0.1234139448	-2.092211169	8.1028120569	2.0922111689
				2.02306730				
				↑ Excelソルバーで最小化				

UNIT Bの累積故障曲線を示す。UNIT Bは、現在量産しており、ROM使用量は1477K, RAM使用量は194 Kである。

実開発のUnit Bの開発変遷と市場保証期間と市場生産台数を以下に記す。開発期間を15か月、各マイルストーン期間を15か月、テストケース3762件、生じた故障数を111件。全故障に対して対策を講じた。また、市場保証期間を10年

市場故障予測分析の結果：

UNIT-Bの算出法別の故障率k, MTBFと市場故障数予測一覧



故障率の 算出方法	テスト完了直 後の故障率 k(件/月)	テスト完了直 後から市場保 証期間10年 間・1000万台 中の市場故障 数予測(台)	同左の 1000万 台中の 無故障 数予測 (台)	テスト完了直後 の平均故障時 間(月) MTBF(Mean Time Between Failure)=1/k	その時点での MTBF>テスト完了 後の市場保証期間 10年完了時(15+10 x 12=135月=580週) となるまでの、テス ト開始からの経過 時間
(1)減衰曲線の 指数関数 近似	0.08件/週 =0.344件/月	10,000,000	0	0.24年=2.9月 =12.6週	2.1年=25.6月=111 週 (テスト後1.0年=11.8 月=51週)
(2)指数形 モデルフィッ ティング	0.033件/週 =0.144件/月	10,000,000	0	0.58年=7.0月 =30.2週	1.8年=21月=91週 (テスト後0.6年=7.2 月=31週)
(3)習熟S字形 モデルフィッ ティング	0.033件/週 =0.144件/月	10,000,000	0	0.58年=7.0月 =30.2週	1.8年=21月=91週 (テスト後0.6年=7.2 月=31週)
(4)ゴンベルツ 曲線モデル フィッティング	0.015件/週 =0.064件/月	9,995,539	4,461	1.3年=15.6月 =67.4週	1.5年=18月=78週 (テスト後0.3年=4.2 月=18週)

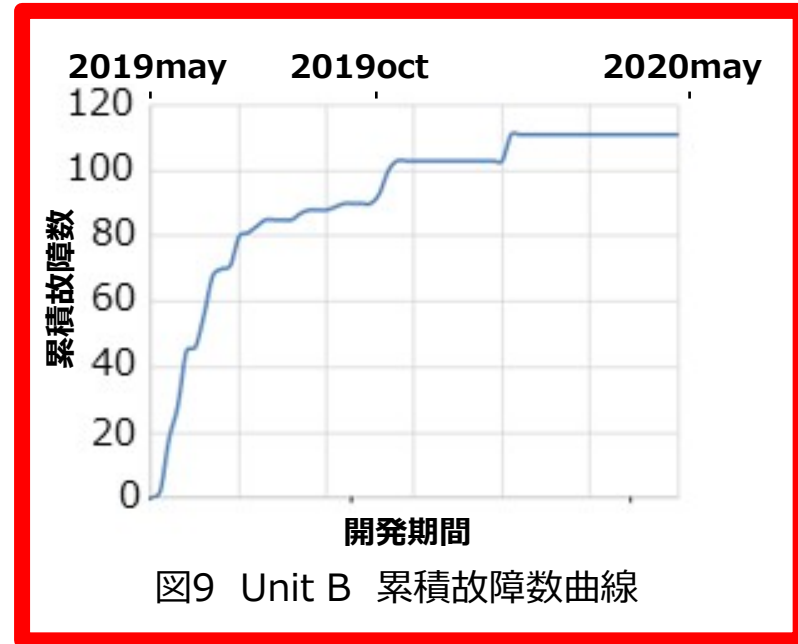


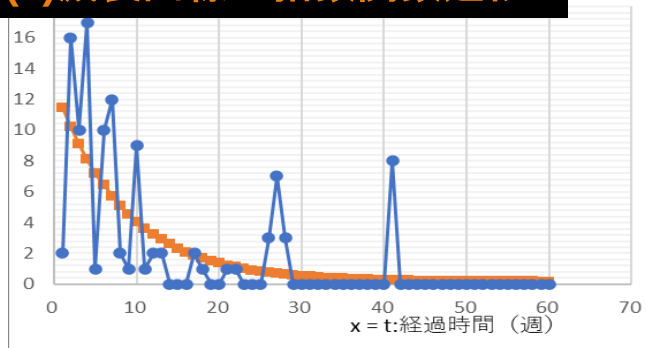
図9 Unit B 累積故障数曲線

補足 追検証データ UNIT B

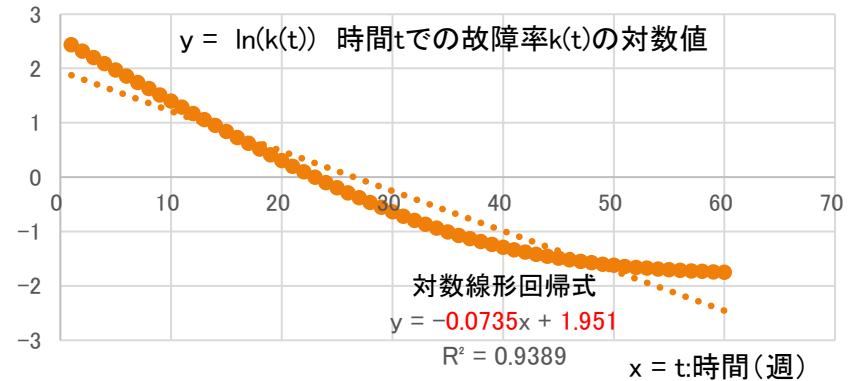
ソフトウェア信頼度成長モデル故障で累積数曲線をカーブフィッティング

(1)減衰曲線の指数関数近似

(補正処)



最小二乗近似の減衰曲線の指数関数近似による経過時間tでの故障率k(t) (対数値) .vs. 経過時間t (週)

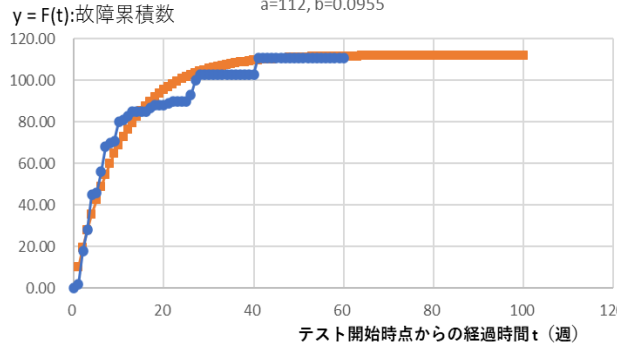


(2)指数形モデル

指数形モデルフィッティング推定曲線 [テスト後も含む推定値]

$$F(t) = a \times (1 - \exp(-b \times t))$$

a=112, b=0.0955

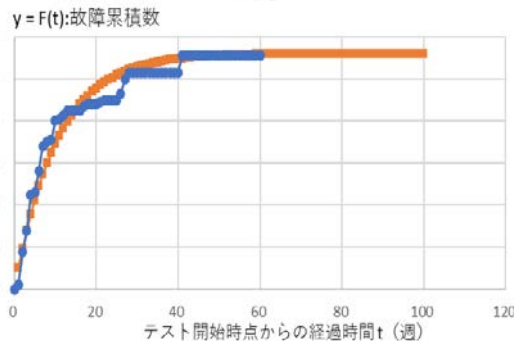


(3)習熟S字形モデル

習熟s字形推定データ

$$F(t) = a \times (1 - \exp(-k \times t))$$

a=112, b=0.0955 [テスト後も含む推定値]

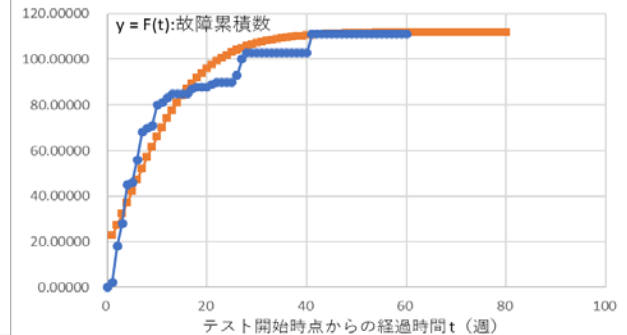


(4)ゴンペルツ曲線モデル

ゴンペルツ曲線フィッティング推定カーブ

$$F(t) = a \times \exp(-b \times e^{-c \times t})$$

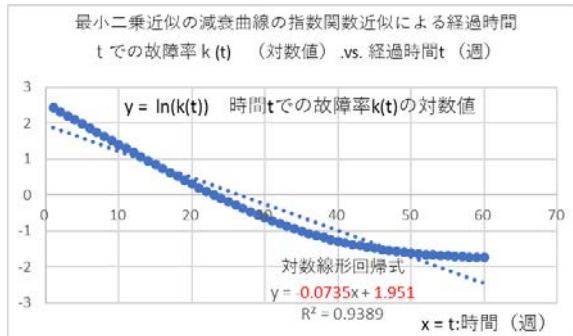
a=112, b=1.80, c=0.1227 [テスト後も含む推定値]



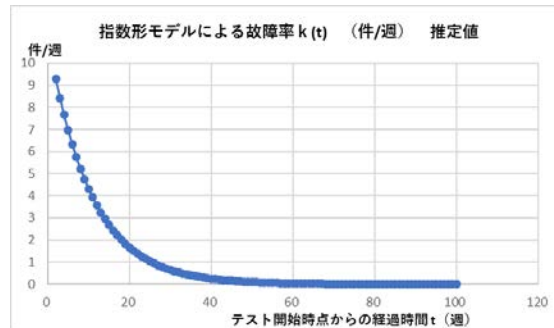
補足 追検証 データ UNIT B

ソフトウェア信頼度成長モデル故障で累積数曲線をカーブフィッティングした推定曲線から故障率を算出

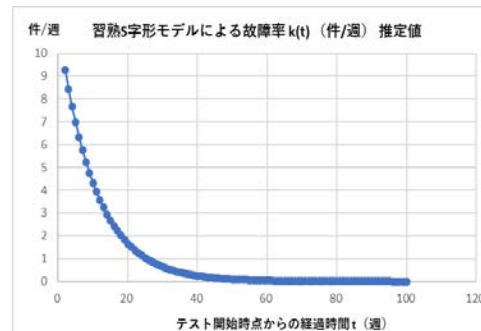
(1)減衰曲線の指数関数近似



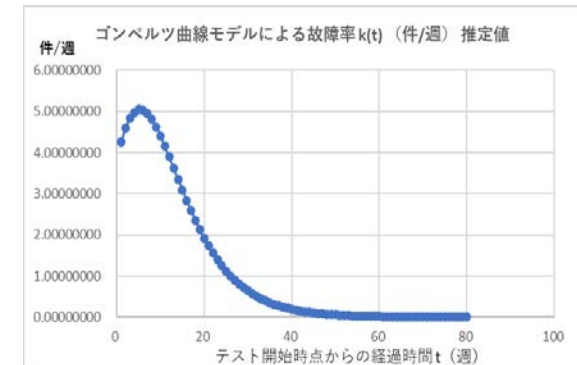
(2)指数形モデル



(3)習熟S字形モデル



(4)ゴンペルツ曲線モデル

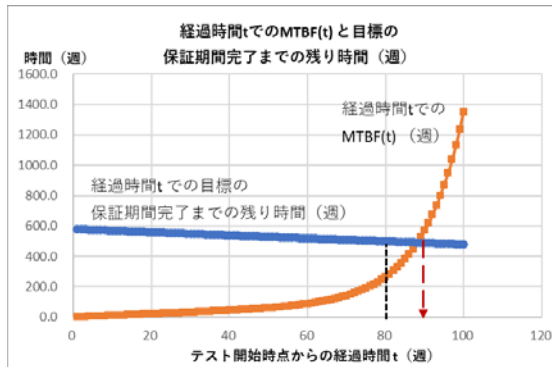


経過時間 t でのMTBF(t)と目標の 保証期間完了までの残り時間(週)

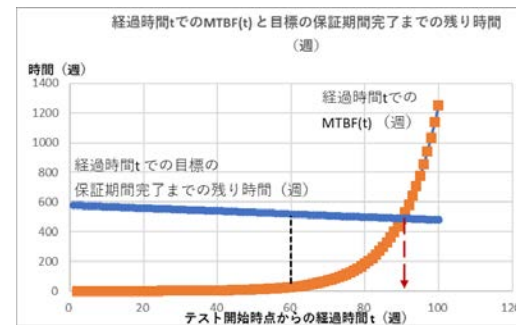
(1)減衰曲線の指数関数近似



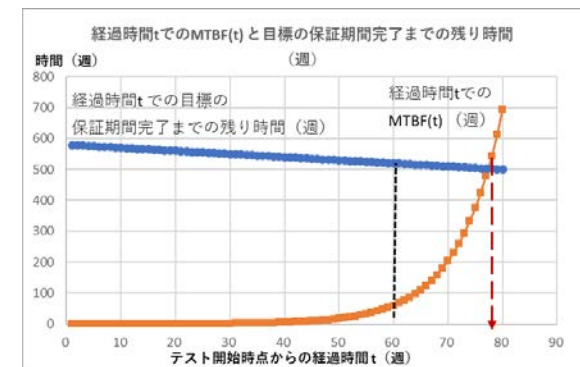
(2)指数形モデル



(3)習熟S字形モデル

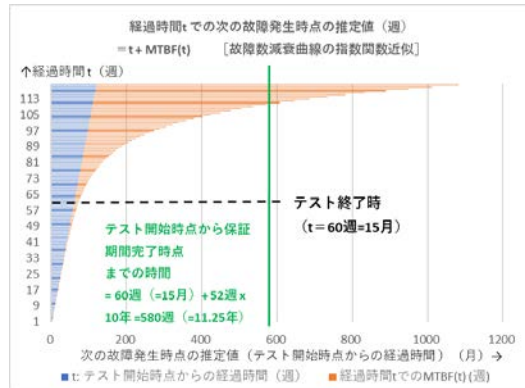


(4)ゴンペルツ曲線モデル

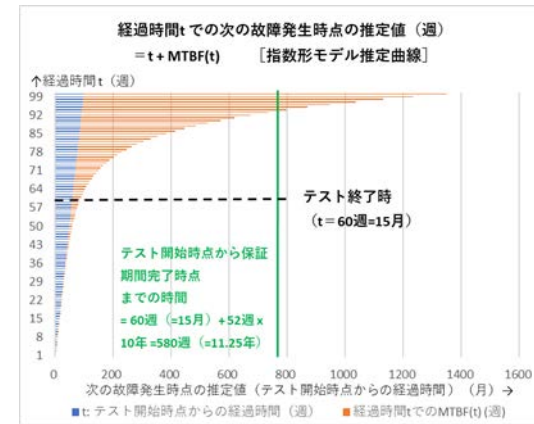


経過時間 t での次の故障発生時点 の推定値(週) $= t + \text{MTBF}(t)$

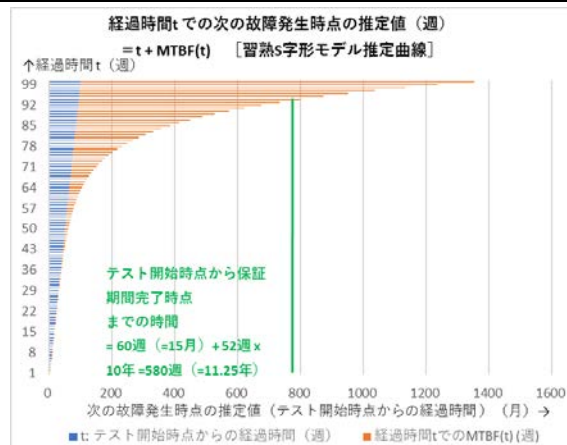
(1)減衰曲線の指数関数近似



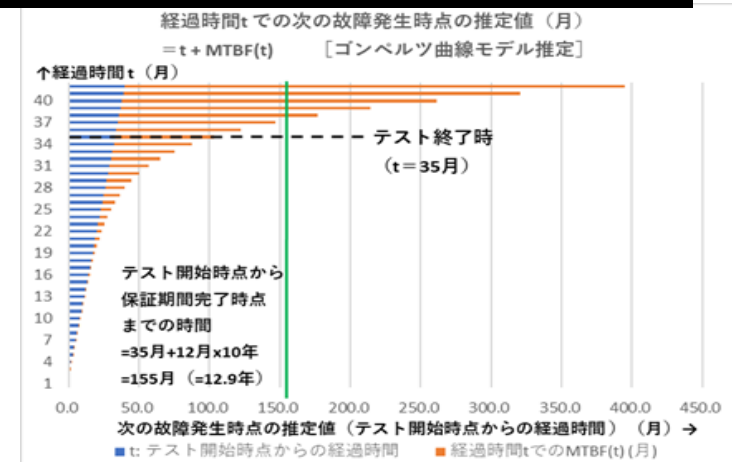
(2)指数形モデルフィッティング



(3)習熟S字形モデルフィッティング



(4)ゴンペルツ曲線モデルフィッティング

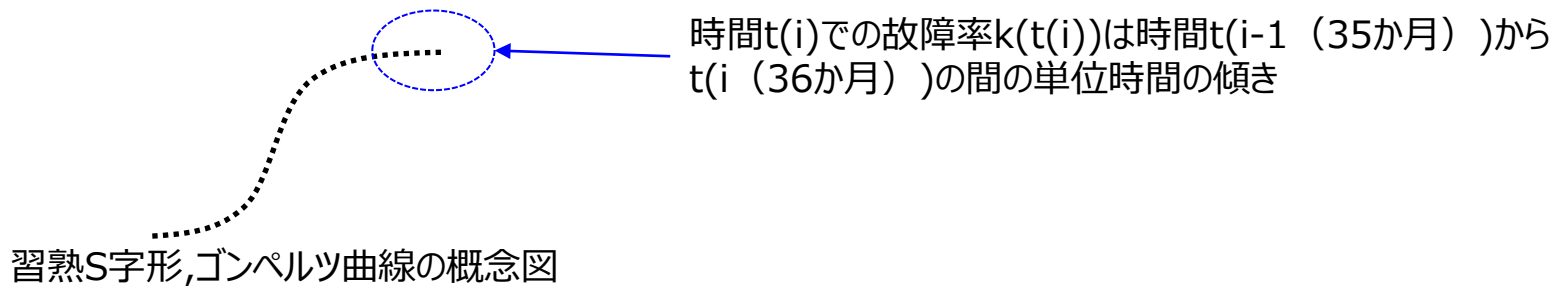


3.5 考察 補足 故障率に関する考察

- 1) カーブフィッティング方式（近似計算）方法によって算出した故障率は、
 (1) 減衰曲線（指数曲線） ≠ (2) 指数形モデル ≠ (3) 習熟S字形 ≠ (4) ゴンペルツ曲線
 (小数点以下1位から3位までの値)

複数の算出法を組合せて一覧し、比較又は選択して用いるのが良いと考える。

- 2) ソフトウェア信頼度成長モデルに関して、故障数が、指数関数的な減衰を示す場合、**S字形モデル（習熟S字形, ゴンペルツ曲線）も指数に非常に近い計算結果となった。** 習熟S字形とゴンペルツ曲線は2点の変曲点を有する特性だが、時間 $t(i)$ での故障率 $k(t(i))$ は時間 $t(i-1)$ （35か月）から $t(i)$ （36か月）の間の単位時間の傾きのためと考える。

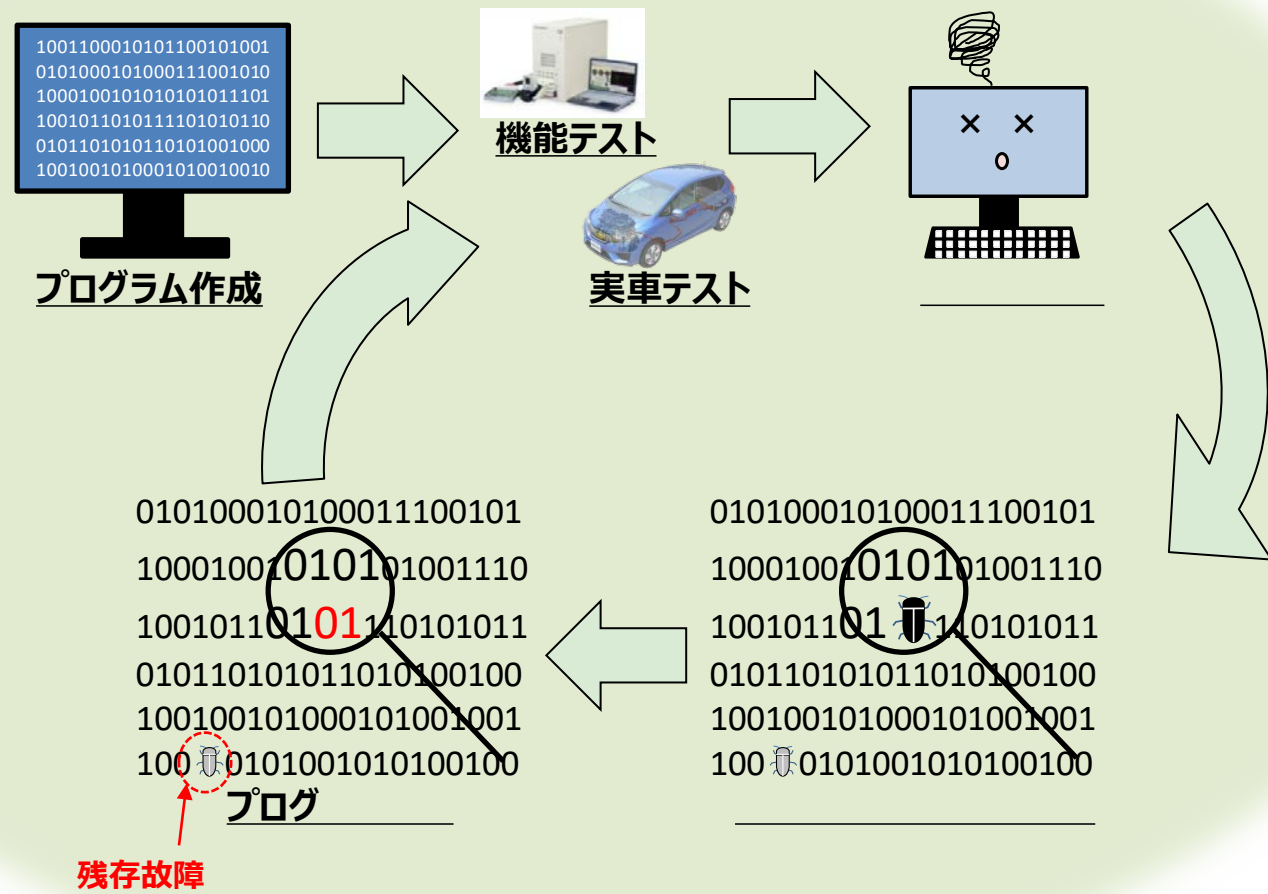


ソフトウェアがリリースされる毎の
受入れテストへの有効性



ソフトウェア開発組織内（取引先様）
のテストへの有効性も期待

2.1 課題



故障を直しても他の故障は残存する可能性があり。
 いつまでテストを実施すればソフトウェアの市場品質を保証できるか？

ソフトウェア受入れテスト時データからの故障率算出

「やったー、故障が出なくなったからテストは終了だね」、「故障をグラフ化することで品質確認できるかな」と検討できる信頼度成長モデルの例

モデル（曲線）	一般式
指数形	$Y = N(1 - e^{-bt})$
ゴンベルツ曲線	$Y = N \exp(-k e^{-bt})$
習熟S字形	$Y = N(1 - e^{-bt}) / (1 + \psi e^{-bt})$

古山恒夫 ソフトウェア信頼度成長モデルに関する統合モデルの解析的パラメータ推定法 情報処理学会論文誌, 1996

テスト終了判断！
残存故障による故障が、市場で生じるのか！
故障が生じる可能性があるならその頻度は？
知りたいのはここだ。



33