C-027

組込みシステム開発のリスクマネジメント

Risk management in embedded system development

小高 文博† 佐藤 建吉‡
Fumihiro Odaka Kenkichi Sato

1．はじめに

組込みシステムは，様々な機器に組み込まれたコンピュータシステムをいい，組込みソフトウェアは，そのコンピュータ上で動作し特定の機能を実現するためのソフトウェアをいう．このソフトウェアは，機器の制御を行いハードウェアと直接的に連携される．本論では，組込みシステムの開発のうち，組込みソフトウェアの開発におけるリスクマネジメントについて述べる．

組込みシステムのほとんどは，いわゆるコンシューマプロダクトと呼ばれる製品群に属しており多くは一般消費者などを対象に大量に出荷される．このため，その出荷後に不具合が発見された場合，その修復には多大の労力とコストがかかる．また，宇宙産業，プラント制御等の特殊な用途向けに個別に開発されるものもあるが，この場合も開発完了後のメンテナンスは困難である．

出荷後の不具合への対応が困難なことは銀行の勘定システムや企業の情報システムのようなエンタプライズ系プロジェクトと全く同様であるが，エンタプライズ系プロジェクトにおいては段階的に機能の運用を開始していくことが可能であり，運用開始後でも機能追加，変更は組込みシステムに比較すると容易である場合が多い．

これらの組込みソフトウェアの開発におけるリスクについてエンタプライズ系ソフトウェアとの開発との違い，性質を明らかにするとともに，リスクに対処する必要なマネジメント方法について論じることは，組込みソフトウェアの開発の効率化，品質向上に寄与するものと考える．

また、リスクを管理するためには、チェックリスト等の管理表が有効であることを示すが、組込みソフトウェアのリスクは、プログラミング段階で顕在化することも多く、プログラミングというプロセスの定義に存在する問題についても言及する。

2．組込み系とエンタプライズ系のソフトウェア開発の違い

ソフトウェア開発を実行する上で，対象となるシステムによりリスクの管理方法も違ったものになるので，それぞれのプロジェクトの特徴を理解しておくことは極めて重要である．

組込みソフトウェアは，軍事・宇宙産業機器から民生機器までさまざまな分野の製品に利用されており，それぞれの製品分野はそれぞれ異なった特徴を持っているが，ここでは大きくエンタプライズ系ソフトウェアとの違いを見ると，①エンタプライズ系のシステムでも開発期間は短くなってきているが，組込みシステムの場合，製品の提供サイクルが短いためエンタプライズ系ソフトウェアに比べその規模に比例せず開発期間は短い．②要求品質に関しては，組込みソフトウェア開発では，設計対象が製品・部品であり品質水準は高く不具合のないものが要求されるが，エンタプライズ向けのソフトウェア開発の場合は，ビジネスに沿ったものであり顧客と合意した品質レベルとなる．出荷量についても，前者が大量であることを前提としているが，後者は，パッケージとして出荷する場合を除き使用範囲は企業内に限られるため少量である．③組込み系技術者の数自体エンタプライズ系に比較し少ないが，各プロジェクトにアサインされる要員数は少ない．しかしその生産性はあまり高くない．④ソフトウェアの製造・試験工程まで完了しても実機に組み込んで動作させる必要があり，ソフトウェアの開発のみで最後の工程まで行なうことはできない．また，ハードウェアとソフトウェアの平行開発になることが多い．⑤ウォーターホール型の開発方法よりスパイラル型の開発方法が採られる傾向が強い．⑥組込みソフトウェアの多くの開発プロジェクトでは，リスクマネジメントを含むプロジェクトマネジメントの重要さが十分に認識されておらず，プロジェクトマネジメントそのものが十分に機能していないといった場合が少なくない．ことがあげられる．

このように組込みソフトウェアではエンタプライズ系のソフトウェアとの違いが多くある．そのためエンタプライズ系のソフトウェア開発と同様なマネジメントは定着しなかったのだろう．組込みソフトウェア開発にリスクマネジメントを定着させるためには，リスクマネジメントを理解し，それぞれの開発対象に応じて適用できるようになる必要がある．

†千葉大学・大学院
‡千葉大学・大学院

3．リスク

一般的にリスクというとマイナスの結果を伴うイメージであるが，経済学や金融工学ではマイナスの側面だけでなく利潤，リターンは大きいが不確実性が高いものも含み，マイナスの結果を必ずしも伴わない．工学的な用語としてリスクは「ある事象生起の確からしさと，それによる負の結果の組み合わせ」（JIS Z 8115:2000）や「事態の確からしさとその結果の組み合わせ，または事態の発生確率とその結果の組み合わせ」（JISQ2001:2001）などと定義されている．本論では，JIS の定義に従い「ある事象の発生確率とその事象の負の結果の組合せ」とする．

また，ソフトウェア開発におけるリスクは，品質に関するリスク，スケジュールに関するリスク，コストに関するリスク等があり，これらのリスクは組込み系システム，エンタプライズ系システムの関わらず潜在する．しかし，組込みソフトウェアの場合，エンタプライズ系ソフトウェアに比較し，前述のようにその開発に特徴があるため，難易が高くリスクが大きいという認識が多いのも事実である．ここで，「表１　プロジェクトにおけるリスク」に筆者が経験した組込みソフトウェアに特有と思われるリスクを示す

表１　プロジェクトにおけるリスク

	PJ 名	リスク	発生したリスクの詳細／影響	対応
1	A	ハードウェアの構成が違う	信号結線図と違う	現地でのソフトウェア修正
2		ハードウェアが動作しない	出荷製品のシリアルインタフェースの結線ミス	結線の変更
3	B	ドライバとの結合時の不具合	ドライバとのタイミングが合わない	ソフトウェア修正
4		既存システムの変更が必要	既存システムのドキュメント類がない	現地でメモリダンプを行い逆アセンブルし修正
5	C	仕様が未確定のまま出荷される	修正が発生（見込済み），修正期間が短い，修正場所が多数	ソフトウェアのパッチ対応

上表を見ると，ソフトウェア開発につきものの“仕様が未確定”や“通信プロトコルの相違”といったリスクの他にハードウェア関連のリスクを想定していることが分かる．組込みソフトウェアは，その名のとおりハードウェアに格納されるため，ハードウェア依存度が高く，一度組み込むと保守が難しくなる．またリソースの制約や処理時間の要求が高い．これは，開発対象がソフトウェアであるにも関わらずハードウェアに関する技術的リスクが大きいことを示している．

4．リスクマネジメント

リスクとは「ある事象の発生確率とその事象の結果の組合せ」を意味すると定義したが，リスクマネジメントは，どのようなリスク要因が存在するのかを調査し，そのリスク要因がどれくらいの頻度で発生するのか想定することである．また，併せてリスク要因によりどの程度の影響を及ぼすか分析評価し，その影響が許容できるか否かを決定し、許容できない場合は必要な対処を行なうプロセスのことである．

ソフトウェア開発の過程においてリスクについて見ると，プロジェクトの進行に伴い，進捗や品質目標に影響を与えるさまざまな事象が発生する可能性がある．こうしたトラブルの多くは事前に予見できる場合があり，プロジェクトの着手前や途中で，どのような潜在的なトラブルの兆しがあり，実際のプロジェクトでどのようなトラブルが発生しているかどうか，また，それらに対して未然に防ぐための対策が講じられているかどうかなどを適切にマネジメントすることが必要である．

また，組込みソフトウェア開発プロセスにおいては，「図１リスクの変化」に示すとおりエンタプライズ系の開発プロセスとは違うリスクの大きさの変化がある．エンタプライズ系の場合は，設計プロセスが終了した時点で，リスクの大きさは大きく減ることになるが，組込み系の場合は，設計プロセスの終了時点でも，リスクはあまり減ることはなく，出荷，運用開始といったプロセスの時点においてもリスクが残ることになる．

具体的なリスクマネジメントの指標として，プロジェクトマネジメント知識体系ガイド（以下 PMBOK）がある．PMBOK は，国際的に標準とされているプロジェクトマネジメントの知識体系で

あり，建設，製造，ソフトウェア開発などを含む幅広いプロジェクトに適用できるプロジェクトマネジメントの基盤となる考え方や手法が提示されているため，リスクマネジメントを含むプロジェクトマネジメントの指標となりうると考える．

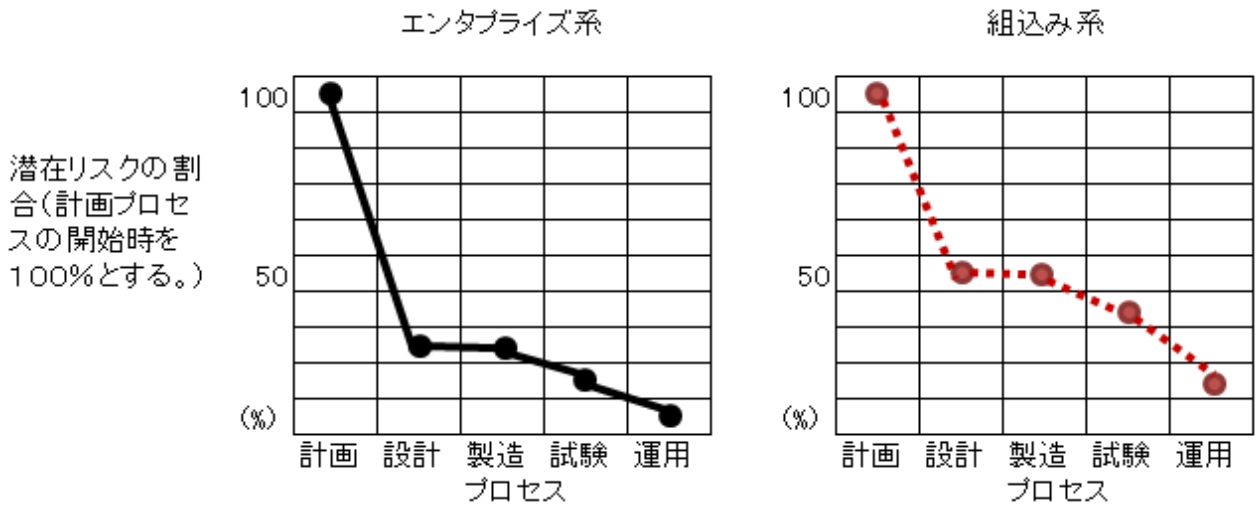

図１　リスクの変化

PMBOK によるとリスクマネジメントは，以下のプロセスを行うことにより達成されることになる．

（１）リスクマネジメント計画を策定する．
（２）プロジェクトのリスク事象とその影響の特定しリスク識別を行う．
（３）定期的なプロジェクト・リスクの見直しを行い，リスクの監視管理を行う．

ここで，PMBOK に従い組込み系ソフトウェア開発で行うリスクマネジメントプロセスを具体的に述べる．

（１）リスクマネジメント計画の策定

プロジェクトのリスクマネジメント活動の計画を立案し基本方針とそれを実行に移すための仕組みを明確にする．これは，リスクのモニタリング方法，軽減策の検討・実施と効果確認といった活動をどう進めていくか検討することが基本となる．

リスクマネジメントの仕組みは，リスクの監視，軽減策の検討・実施・評価といった活動をどのような実施方法でどのようなタイミングで進めるかを明確にする．記述の際に注意考慮すべき内容は，リスクの監視や洗い出しについては，プロジェクトの将来を見通したリスクの予見を行うことである．

また，仕組みを実施する体制については，プロジェクト内で誰がリスクマネジメントを主体的に進めていくかを明確にする必要がある．

（２）プロジェクトのリスク識別

プロジェクトにおいて発生が予見されるさまざまなリスクを洗い出し，後述するリスク管理表として整理する．洗い出したリスクを整理し，それらに対する軽減策やその実施状況を整理し，リスクの状況を常に見えるようにしておくことが重要である．リスクマネジメントのタイミングを逸しないためにリスク管理表が有効である．

リスク管理表に記載すべき項目および記述内容を示す．①リスクのカテゴリ（製品規模，ユーザ特性，ビジネス特性，プロセス，技術，開発環境，リソース），②リスクの内容（リスクが次工程に与える影響，プロジェクト全体に与える影響，インパクト（発生頻度，リスク重要度）），③対応策，④リスク検出日，⑤対策完了日，⑥滞留日，⑦対策担当である．

また，リスクの洗い出しと評価ができるよう，洗い出されたリスクについては，そのリスクが発生する確率や可能性などを検討する．実際にリスクが開発のどの時点で発生し，そのリスクによるプロジェクトに対する影響がどの程度かも合わせて評価する．リスクの重要度や発現の可能性を考慮して，リスク軽減のための対策を検討し整理する．合わせて軽減策実施の期限や責任者なども明確にしておく．

ソフトウェアプロジェクトで発現するリスクには技術面のリスク，リソース面のリスクなどいくつかのカテゴリで分類することができる．

リスクを洗い出す時には問題となりそうな現象をリスクとしてとらえるだけでなく，その原因を洗い出す必要がある．リスクの根源に対策を打つことで，単一の問題だけでなく，そのリスク要因を分類することにより，リスク特定時の発見漏れを少なくすることができる．

（３）定期的なリスクの監視管理

リスクはプロジェクトの進行に伴い，さまざまな新規リスクが発生したり，既に予見できているリスクがさらに別のリスクを誘引したりする場合があるため，随時，リスク管理表の見直しを行う．

その場合には、ＰＤＣＡのサイクルに従いリスクマネジメントを行なうことが基本となる．

5．実務における対応

ある組込みソフトウェアの開発をとおして，前述のリスクマネジメントを適用し具体的に発生したリスクについて以下にリスク，対応策，実際の対応を述べる．

（１）ハードウェア開発が遅延する．

対応策：ハードウェア開発が遅れた場合のソフトウェア開発側の対策を検討する．

実際の対応：実機の提供が遅れそうになったため，クリティカルパス上の作業だけでも進められるような試作機をリリースしてもらうなど交渉した．

（２）性能問題が発生する．

対応策：目標性能など重要な要件については，対処できるレベルまでモジュールを分解し調査する．

実際の対応：ハードウェアの性能上の問題が発生した場合，処理の遷移の途中に確認画面を表示させるなどユーザのオペレーションを介在させることにより体感する処理時間を短く感じさせることにした．

（３）リソース不足が起こる．

対応策：メモリ容量など使用できる資源に限度がある場合は，段階的に対処していくことになるので解消されるまで継続して管理する．

実際の対応：複数処理を同時に行う場合，メモリネックとなりドライバがロードできないときがあった．これは最終的にコードの見直しを行い適切なサイズにすることができたが場当たり的な対処にならず段階的にモジュールサイズの変更が行なえた．

特に（２），（３）に関してはリスク管理を行うことにより一次リスク（性能問題）から二次リスク（技術的な問題）の発見を行うことができ，結果的に技術的な問題，つまり技術力の向上により解決できる問題に繋げることができたと考える．

6．まとめ

組込みソフトウェアの開発のリスク管理は、最終形態であるプログラミングを作る技術（コーディング）力の問題である。多くのリスク管理の問題は、プログラミング時の問題であるということを意味する。しかし、リスクは、上流工程の要件定義や設計に潜在しているものであり、後工程である製造プロセスにほとんど潜在しないはずである。しかし、ソフトウェア開発では、コーディング時にプログラマは要件定義レベルにまで遡り、上位レベルの設計をやり直し、それに従ってプログラミングを行うこともあることがしばしばあることも事実である。プログラミングの位置づけそのものに、ソフトウェア開発のリスクの根本があるのではないかと考えられる。プログラミングは、「設計書に基づきコードを書くという製造プロセス」ではなく、「ソフトウェア開発の詳細設計工程のプロセス」と考え、上流工程と同様のプロセスやリスクの管理を行うことが必要である。

参考文献

[1] Project Management Institute，プロジェクトマネジメント知識体系ガイド第３版，PMI，2004

[2] 独立行政法人 IPA/SEC 編，組込みソフトウェア向けプロジェクトマネジメントガイド［計画書編］，翔泳社，2006

[3] 独立行政法人ＩＰＡ/SEC 編著，組込みソフトウェア開発向け品質作り込みガイド，翔泳社，2008

[4] プロジェクトマネジメント学会，リスク管理表を活用したプロジェクト・リスク・メンジメント，プロジェクトマネジメント学会，2008