目次

第 1 章　安全ライフサイクルにおけるシステムエンジニアリング.....8

1.1　システムエンジニアリングの役割.....9

1.1.1　システムエンジニアリングの目的.....9

1.1.2　システムエンジニアリングの活動.....10

1.2　安全ライフサイクルにおけるシステムエンジニアリング.....11

1.2.1　階層的なシステムエンジニアリング.....11

1.2.2　車両レベルのシステムエンジニアリングにおける安全活動.....11

1.2.3　システムレベルのシステムエンジニアリングにおける安全活動.....13

第 2 章　アイテム定義.....14

2.1　アイテム定義の概要.....15

2.1.1　アイテム定義の目的.....15

2.1.2　アイテム定義の活動.....16

2.2　アイテム定義の詳細解説.....17

2.2.1　車両機能の定義.....17

2.2.2　ユースケース分析.....17

2.2.3　システムコンテキスト分析.....20

2.2.4　文書化（製品の要件定義）.....23

2.2.5　検証レビュー.....23

セルフチェックリスト：アイテム定義.....24

第 3 章　ハザード分析とリスクアセスメント.....26

3.1　ハザード分析とリスクアセスメント　概要.....27

3.1.1　ハザード分析とリスクアセスメントの目的.....27

3.1.2　ハザード分析とリスクアセスメントの活動.....27

3.2　ハザード分析とリスクアセスメント　詳細解説.....28

3.2.1　ハザードの識別.....28

3.2.2　運用状況の分析.....30

3.2.3　危険事象の決定.....31

3.2.4　事故シナリオの特定.....31

3.2.5　危害の大きさの見積もり.....31

3.2.6　制御可能性の見積もり.....32

3.2.7 リスクの評価 34
3.2.8 安全目標 35
3.2.9 ハザード分析及びリスクアセスメントの検証レビュー 36
セルフチェックリスト：ハザード分析とリスクアセスメント 37

第 4 章 機能安全コンセプト 40

4.1 機能安全コンセプトの概要 41
4.1.1 機能安全コンセプト開発の目的 41
4.1.2 機能安全コンセプト開発の活動 41
4.2 機能安全コンセプトの詳細解説 42
4.2.1 安全状態の定義 42
4.2.2 安全構想設計 44
4.2.3 機能安全要件定義 47
4.2.4 車両レベルのアーキテクチャ設計 49
4.2.5 機能安全要件の割り当て 53
4.2.6 妥当性確認の基準定義 53
4.2.7 リスクアセスメント結果及び機能安全コンセプトの検証レビュー 55
セルフチェックリスト：機能安全コンセプト 56

第 5 章 技術安全コンセプト 58

5.1 技術安全コンセプトの概要 59
5.1.1 技術安全コンセプト開発の目的 59
5.1.2 技術安全コンセプト開発の活動 59
5.2 技術安全コンセプトの詳細解説 60
5.2.1 技術安全要件定義 60
5.2.2 安全機構設計 62
5.2.3 システムアーキテクチャ設計 65
5.2.4 システムアーキテクチャの評価 72
5.2.5 技術安全コンセプトの検証レビュー 75
セルフチェックリスト：技術安全コンセプト 76

第 6 章 システム・アイテム統合とテスト 78

6.1 システム・アイテム統合とテストの概要 79
6.1.1 システム・アイテム統合とテストの目的 79
6.1.2 システム・アイテム統合とテストの活動 79
6.2 システム・アイテム統合とテストの詳細解説 80

6.2.1 統合戦略の立案 80
6.2.2 テスト戦略の立案 80
6.2.3 テスト設計 81
6.2.4 統合及び統合テストの実施 89
セルフチェックリスト：システム・アイテム統合とテスト 90
第 7 章 安全妥当性確認 92
7.1 安全妥当性確認の概要 93
7.1.1 安全妥当性確認の目的 93
7.1.2 安全妥当性確認の活動 93
7.2 安全妥当性確認の詳細解説 94
7.2.1 安全妥当性確認戦略の立案 94
7.2.2 安全妥当性確認テスト設計 94
7.2.3 安全妥当性確認の実施 95
7.2.4 安全妥当性確認結果の評価 95
セルフチェックリスト：安全妥当性確認 96
第 8 章 【共通エンジニアリング】安全要件管理 98
8.1 安全要件定義 99
8.1.1 安全要件定義の目的 99
8.1.2 安全要件記述に対する要求事項 99
8.2 安全要件の管理 102
8.2.1 安全要件管理の目的 102
8.2.2 安全要件管理への要求事項 102
第 9 章 【共通エンジニアリング】安全分析 104
9.1 安全分析 105
9.1.1 安全分析の目的 105
9.1.2 安全分析の技法 106
9.2 従属故障分析 112
9.2.1 従属故障分析の目的 112
9.2.2 従属故障分析の手順 112
9.2.3 カップリングファクタ識別の考慮点 115
第 10 章 【共通エンジニアリング】ASIL 依存のエンジニアリング 118
10.1 ASIL デコンポジション 119

10.1.1 ASIL デコンポジションの目的 119
10.1.2 ASIL デコンポジション適用における考慮点 120
10.2 異なる ASIL エレメントの共存 121
10.2.1 異なる ASIL エレメントの共存　活動の目的 121
10.2.2 異なる ASIL エレメントを共存させるための考慮点 122
第 11 章 【共通エンジニアリング】既存エレメントの利用 124
11.1 既存エレメントの再利用 125
11.1.1 既存エレメント再利用の目的 125
11.1.2 既存エレメント開発における役割 125
11.2 影響分析 126
11.3 安全関連系への COTS 利用 127
11.3.1 ソフトウエア COTS の利用 127
11.3.2 ハードウエア COTS の利用 128
11.4 安全関連系への SEooC 利用 129
11.4.1 SEooC 開発の流れ 129
11.4.2 想定の開発と利用の評価 130
11.4.3 SEooC の統合と評価 130

目次

第 1 章　安全ライフサイクルにおけるハードウエアエンジニアリング 6

1.1　ハードウエアエンジニアリングの役割 7

1.1.1　ハードウエアエンジニアリングの目的 7

1.1.2　ハードウエアエンジニアリングの活動 8

1.2　安全ライフサイクルにおけるハードウエアエンジニアリング 9

1.2.1　ハードウエアエンジニアリングにおける安全活動 9

第 2 章　ハードウエア安全要求の仕様化 10

2.1　ハードウエア安全要求の仕様化の概要 11

2.1.1　ハードウエア安全要求仕様化の目的 11

2.1.2　ハードウエア安全要求仕様化の活動 11

2.2　ハードウエア安全要求仕様化の詳細解説 12

2.2.1　ハードウエア安全要求の定義 12

2.2.2　ハードウエアとソフトウエア間のインタフェース仕様（HSI）の更新 16

2.2.3　ハードウエア安全要求の検証 16

2.2.4　ハードウエアとソフトウエア間のインタフェース仕様（HSI）の検証 16

セルフチェックリスト：ハードウエア安全要求の仕様化 17

第 3 章　ハードウエア設計 18

3.1　ハードウエア設計 19

3.1.1　ハードウエア設計の目的 19

3.1.2　ハードウエア設計の活動 19

3.2　ハードウエア設計の詳細解説 21

3.2.1　ハードウエアアーキテクチャ設計 21

3.2.2　ハードウエア詳細設計 22

3.2.3　ハードウエア設計の検証 23

セルフチェックリスト：ハードウエア設計 26

第 4 章　ハードウエアアーキテクチャ設計の評価 28

4.1　ハードウエアアーキテクチャメトリック評価の概要 29

4.1.1　ハードウエアアーキテクチャメトリック評価の目的 29

4.1.2　ハードウエアアーキテクチャメトリック評価の活動 29

4.2 ハードウエアアーキテクチャメトリック評価の詳細解説 30
4.2.1 フォールトと故障 30
4.2.2 シングルポイントフォールトメトリック（SPFM）の計算 31
4.2.3 潜在フォールトメトリック（LFM）の計算 33
4.2.4 ハードウエアアーキテクチャメトリックの評価手順 34
4.2.5 ハードウエアアーキテクチャメトリック評価結果の検証レビュー 39
セルフチェックリスト：ハードウエアアーキテクチャ設計の評価 40
第 5 章 ランダムハードウエア故障による安全目標侵害の評価 42
5.1 ランダムハードウエア故障による安全目標侵害評価の概要 43
5.1.1 ランダムハードウエア故障による安全目標侵害評価の目的 43
5.1.2 ランダムハードウエア故障による安全目標侵害評価の活動 43
5.2 ランダムハードウエア故障による安全目標侵害評価の詳細解説 44
5.2.1 評価手法の選択 44
5.2.2 安全目標侵害確率の評価（PMHF） 44
5.2.3 安全目標侵害の各原因の評価（EEC） 48
5.2.4 検証レビューの実施 50
セルフチェックリスト：ランダムハードウエア故障による安全目標侵害の評価 51
第 6 章 ハードウエア統合と統合テスト 52
6.1 ハードウエア統合と統合テストの概要 53
6.1.1 ハードウエア統合と統合テストの目的 53
6.1.2 ハードウエア統合と統合テストの活動 53
6.2 ハードウエア統合と統合テストの詳細解説 54
6.2.1 ハードウエア統合テストの計画 54
6.2.2 ハードウエア統合テストの設計 54
6.2.3 ハードウエア統合及び統合テストの実施 59
セルフチェックリスト：ハードウエア統合と統合テスト 60
第 7 章 【共通エンジニアリング】安全要求管理 62
7.1 安全要求定義 63
7.1.1 安全要求定義の目的 63
7.1.2 安全要求記述に対する要求事項 63
7.2 安全要求の管理 66
7.2.1 安全要求管理の目的 66
7.2.2 安全要求管理への要求事項 66

第 8 章　【共通エンジニアリング】安全分析 68
8.1　安全分析 69
8.1.1　安全分析の目的 69
8.1.2　安全分析の技法 70
8.2　従属故障分析 76
8.2.1　従属故障分析の目的 76
8.2.2　従属故障分析の手順 76
8.2.3　カップリングファクタ識別の考慮点 79
第 9 章　【共通エンジニアリング】ASIL 依存のエンジニアリング 82
9.1　ASIL デコンポジション 83
9.1.1　ASIL デコンポジションの目的 83
9.1.2　ASIL デコンポジション適用における考慮点 84
9.2　異なる ASIL エレメントの共存 85
9.2.1　異なる ASIL エレメントの共存　活動の目的 85
9.2.2　異なる ASIL エレメントを共存させるための考慮点 86
第 10 章　【共通エンジニアリング】既存エレメントの利用 88
10.1　既存エレメントの再利用 89
10.1.1　既存エレメント再利用の目的 89
10.1.2　既存エレメント開発における役割 89
10.2　影響分析 90
10.3　安全関連系への COTS 利用 91
10.3.1　ソフトウエア COTS の利用 91
10.3.2　ハードウエア COTS の利用 92
10.4　安全関連系への SEooC 利用 93
10.4.1　SEooC 開発の流れ 93
10.4.2　想定の開発と利用の評価 94
10.4.3　SEooC の統合と評価 94

目次

第 1 章　安全ライフサイクルにおけるソフトウエアエンジニアリング 8

1.1　ソフトウエアエンジニアリングの役割 9

1.1.1　ソフトウエアエンジニアリングの目的 9

1.1.2　ソフトウエアエンジニアリングの活動 10

1.2　安全ライフサイクルにおけるソフトウエアエンジニアリング 11

1.2.1　ソフトウエアレベルにおける安全活動 11

第 2 章　ソフトウエア安全要求の仕様化 14

2.1　ソフトウエア安全要求の仕様化の概要 15

2.1.1　ソフトウエア安全要求の仕様化の目的 15

2.1.2　ソフトウエア安全要求の仕様化の活動 15

2.2　ソフトウエア安全要求の仕様化の詳細解説 16

2.2.1　ソフトウエア安全要求の定義 16

2.2.2　ハードウエアとソフトウエア間のインタフェース仕様（HSI）の更新 18

2.2.3　ソフトウエア安全要求の検証 18

2.2.4　HSI の検証 18

セルフチェックリスト：ソフトウエア安全要求の仕様化 19

第 3 章　ソフトウエアアーキテクチャ設計 20

3.1　ソフトウエアアーキテクチャ設計の概要 21

3.1.1　ソフトウエアアーキテクチャ設計の目的 21

3.1.2　ソフトウエアアーキテクチャ設計の活動 21

3.2　ソフトウエアアーキテクチャ設計の詳細解説 22

3.2.1　ソフトウエアアーキテクチャ設計 22

3.2.2　安全分析 29

3.2.3　ソフトウエアアーキテクチャ設計の検証 31

セルフチェックリスト：ソフトウエアアーキテクチャ設計 33

第 4 章　ソフトウエアユニット設計及び実装 34

4.1　ソフトウエアユニット設計及び実装の概要 35

4.1.1　ソフトウエアユニット設計及び実装の目的 35

4.1.2　ソフトウエアユニット設計及び実装の活動 35

4.2 ソフトウエアユニット設計及び実装の詳細解説 36
4.2.1 ソフトウエアユニット設計 36
4.2.2 ソフトウエアユニットの実装 37
セルフチェックリスト：ソフトウエアユニット設計及び実装 38
第 5 章 ソフトウエアユニット検証 40
5.1 ソフトウエアユニット検証の概要 41
5.1.1 ソフトウエアユニット検証の目的 41
5.1.2 ソフトウエアユニット検証の活動 41
5.2 ソフトウエアユニット検証の詳細解説 42
5.2.1 ソフトウエアユニット検証戦略の立案 42
5.2.2 ソフトウエアユニットの静的検証 42
5.2.3 ソフトウエアユニットテストの設計 44
5.2.4 ソフトウエアユニットテストの実施 48
セルフチェックリスト：ソフトウエアユニット検証 49
第 6 章 ソフトウエア統合及び検証 50
6.1 ソフトウエア統合及び検証の概要 51
6.1.1 ソフトウエア統合及び検証の目的 51
6.1.2 ソフトウエア統合及び検証の活動 51
6.2 ソフトウエア統合及び検証の詳細解説 52
6.2.1 ソフトウエア統合戦略の立案 52
6.2.2 ソフトウエア統合テスト戦略の立案 52
6.2.3 ソフトウエア統合テストの設計 52
6.2.4 ソフトウエア統合及び統合テストの実施 56
セルフチェックリスト：ソフトウエア統合及び検証 57
第 7 章 組込みソフトウエアのテスト 58
7.1 組込みソフトウエアのテストの概要 59
7.1.1 組込みソフトウエアのテストの目的 59
7.1.2 組込みソフトウエアのテストの活動 59
7.2 組込みソフトウエアのテストの詳細解説 60
7.2.1 組込みソフトウエアテスト戦略の立案 60
7.2.2 組込みソフトウエアテストの設計 60
7.2.3 組込みソフトウエアテストの実施 63
セルフチェックリスト：組込みソフトウエアのテスト 64

第 8 章　【共通エンジニアリング】安全要件管理 66
8.1　安全要件定義 67
8.1.1　安全要件定義の目的 67
8.1.2　安全要件記述に対する要求事項 67
8.2　安全要件の管理 70
8.2.1　安全要件管理の目的 70
8.2.2　安全要件管理への要求事項 70
第 9 章　【共通エンジニアリング】安全分析 72
9.1　安全分析 73
9.1.1　安全分析の目的 73
9.1.2　安全分析の技法 74
9.2　従属故障分析 80
9.2.1　従属故障分析の目的 80
9.2.2　従属故障分析の手順 80
9.2.3　カップリングファクタ識別の考慮点 83
第 10 章　【共通エンジニアリング】ASIL 依存のエンジニアリング 86
10.1　ASIL デコンポジション 87
10.1.1　ASIL デコンポジションの目的 87
10.1.2　ASIL デコンポジション適用における考慮点 88
10.2　異なる ASIL エレメントの共存 89
10.2.1　異なる ASIL エレメントの共存　活動の目的 89
10.2.2　異なる ASIL エレメントを共存させるための考慮点 90
第 11 章　【共通エンジニアリング】既存エレメントの利用 92
11.1　既存エレメントの再利用 93
11.1.1　既存エレメント再利用の目的 93
11.1.2　既存エレメント開発における役割 93
11.2　影響分析 94
11.3　安全関連系への COTS 利用 95
11.3.1　ソフトウエア COTS の利用 95
11.3.2　ハードウエア COTS の利用 96
11.4　安全関連系への SEooC 利用 97
11.4.1　SEooC 開発の流れ 97

11.4.2 想定の開発と利用の評価 98
11.4.3 SEooC の統合と評価 98