発表用の台本

P1　これから矢吹研究室，工藤亮の発表を始めます．

私の研究テーマはプロジェクトホスティングサービスのためのEVM自動描画システムの開発です．

P2　発表内容はこのように進めます．

P3　まず，背景としまして，現在のソフトウェア開発には，バージョン管理ツールが必要です．バージョン管理ツールをホスティングするサービスのことをホスティングサービスといいます．

P4　ホスティングサービスには，ギットハブやアリオスなどのいろいろな種類があります．ホスティングサービスの種類の中でもギットハブが代表的なツールとされています．プロジェクトマネジメントを行う人は，PMBOKで記述されているツールを使用したいので，実際にGitHubを使用し，PMBOKに記述されているツールが使用できるかを調査しました．その結果，PMBOKに記述されているツールの中でもプロジェクトの進捗管理で重要なEVMはGitHubではサポートされていませんでした．

P5　その結果から，PMBOKに記述されているツールの中でも重要なEVMをGitHubの機能を利用し作成後，GitHub上に表示するシステムの開発を目的とします．

P6　目的として行うシステムができたとして，そのシステムの運用方法はこのようになっています．プロジェクトマネジャーは必要なタスクの分だけIssuesを発行します．このIssuesとは，GitHub内でタスク管理をする機能です．そのIssuesには，このあとに説明する手法①のEVMのために必要な情報を記述します．そして，プロジェクトマネジャーかメンバーはそのタスクのIssuesに対応した進捗報告をcommentとして返信します．このcommentとは，Issuesに対する記述ができる箇所のことです．この運用方法によって，Issuesとcommentの進捗報告の内容でEVMを作成する情報をそろえることができます．

P7　手法はこのように進めます．まず，プロジェクトマネジャーが「EVMのために必要な情報」をGitHub内に入力し，「その情報をプログラムによって取り出し」，取り出した情報から「EVMを描画」します．そして「描画したEVMをGitHub上にうめこむこと」で目的が達成できると考えます．

次に各手法について説明します．  
P8　手法1のEVMのために必要な情報について説明します．まず，EVMのために必要な情報を記述するのに使用する機能は，システムの運用方法で指定したとおりにIssuesとcommentに記述します．GitHubには，記述できる機能は，WikiとIssuesの2つがあります．wikiは文章を入力するだけの機能ですが，Issuesはタスク管理をする機能なので，Issuesを使用することに決めます．

P9　そして，Issuesの記述内容は，「タスク名」「タスク期限日」「計画予定時間」「時給金額」「材料費」です．この情報がEVMのために必要な情報です．Issuesを発行する場合は，この内容を記述します  
P10　次に発行されたIssuesに対応するコメントの記述内容は「直接労働時間数」「進捗測定基準度」「使用材料費」です．さらに，「Issuesにコメントした時刻」を加えることによってコメントからのEVMのために必要な情報をそろえることができます．「Issuesにコメントした時刻」を記述しない理由は，commentしたときにGitHub上に表示されるので，記述する必要はないと判断します.

これらのIssusとcommentによって，EVMのために必要な情報をそろえることができると考えます．次にこれらの情報をGitHubから取り出す手法について説明します．  
P11　ここでは，EVMのために必要な情報をGitHubから取り出す手法について説明します．GitHubから取り出す際に使用するツールはGitHub APIです．GitHub APIはGitHubをプログラムから操作するときに必要な，利用するインターフェースです．使用するGitHub APIは次の2つを使用します．  
P12　使用するGitHub APIはこの2つです．APIの説明のリポジトリとは，GitHub上の1つのプロジェクトを管理する単位です．上に記述したAPIによって取り出す情報は，発行したIssuesに記述した，「タスク名」「タスク期限日」「計画予定時間」「時給金額」「材料費」を取り出します．次に下に表示したAPIで取り出せる情報は「直接労働時間数」「進捗測定基準度」「使用材料費」「Issuesにcommentした時刻」です．この取り出した情報によってEVMを描画します．次に取得した情報からEVMを描画する手法について説明します．

P13　描画するときに使用するツールは，Google Chart APIです．Google Chart APIは，書きたいグラフに対応したURLを作成すると，描画してくれるサービスです．取り出した情報をもとにURLを作成することで，EVMを描画すると考えます．次に描画したEVMをGitHub上に挿入する手法について説明します．

P14　GitHub上に描画したEVMをうめこむときに使用するツールは，ブックマークレットです．ブックマークレットは，ブックマークにJavaScriptによる特定の動作を仕組んだものです．このブックマークレットを使用することによってGitHub上にEVMをうめこむことができると考えます．これまでに紹介した手法をもとにGitHub上にEVMを表示するシステムの構築を行いました．次に，構築したシステムの動作確認について説明します．

P15　動作確認はこのような順番で進行します．まず，仮想プロジェクトデータをIssuesに登録し，登録した内容から必要な情報を取り出すことができるか「GitHub APIの動作確認」をします．次にその情報からEVMを作成できるか「EVM作成の動作確認」をします．そして，作成されたEVMがGitHub上に埋め込めるか「GitHub上にEVMを表示する動作確認」をします．動作確認は実際に進行するプロジェクトのデータを利用した場合にシステムに不具合がないかを確認するために実際にシステムを動かし使用することで確認します．この動作確認では，それぞれの動作が正常に稼働するかを確認します．

P16　これは，動作確認で使用する仮想プロジェクトデータです．動作確認をするために作成したリポジトリで，システムの運用方法に沿ってタスクのデータを入力し，Issuesによって指示し，そのIssuesに対応した内容をコメントに記述することによって，進行するプロジェクトのタスクの管理を再現したデータです．

P17　ここでは，GitHub APIでEVMに必要な情報を取り出す動作確認をします．GitHub APIによって仮想プロジェクトデータのIssuesとcomment内容を取り出すことができるか確認します．確認した結果，EVMのための情報を正常に取り出すことができました． P18　ここでは，EVM作成の動作確認をします．動作確認には，仮想プロジェクトデータのEVMイメージ図を使用します．仮想プロジェクトデータから，EVMを作成し，描画されるEVMをExcelで描画し，システムで描画されたEVMが同じように，描画できているか確認をします．描画されている図を見比べると，同じなので，このEVM作成の動作確認は正常であると判断できます．次にGitHub上でのEVM表示の動作確認をします．  
P19　このスライドに表示されている画像は，GitHubの標準の画面です．この動作確認は，GitHub上にEVM表示することができるか判断します．

P20　これは，GitHub上にEVM表示することができた状態の画像です．この画像から，EVMがGitHub上に挿入できると判断できます．

P21　これまでの動作確認によって，システムで使用するそれぞれの動作は，正常に稼働したことが確認できました．この結果からシステムに不具合がないと判断でき，GitHubのIssuesから必要な状態を取り出し，GitHub上にEVMを表示することが可能です．

P22　まとめとしては，GitHubは標準ではEVMを描くことはできません．そこで，GitHubでEVMを描けるようにしたいと考えました．描くには，Issuesとcommentの使い方をきめて，その内容をGitHub APIで取り出せるようにします．そして，取り出した情報をGoogle Chart APIでグラフを描けるようにし，ブックマークレットでGitHubにグラフをうめこめるようにします．このようにすれば，GitHub上にEVMを描くことができると考え，システムを構築しました．そして，実際にシステムを動かし正常に動作できることが確認できました．

P23　このシステムによってソフトウェア開発の現場でよく使用されているGitHubにPMBOKで言われているようなプロジェクトマネジメントのツールを導入することができるようになりました．

以上で発表を終わりにします．ご清聴ありがとうございます．