

# プロジェクト管理(C++)

## ガントチャートの使い方

スケジュールを立てても今の自分達がどこらへんまで進んでいるのか、特に複数人で開発している場合はどこが遅れているのか、誰が遅れているのかを正確に把握する必要があります。

そこで便利なのがガントチャートというツールです。

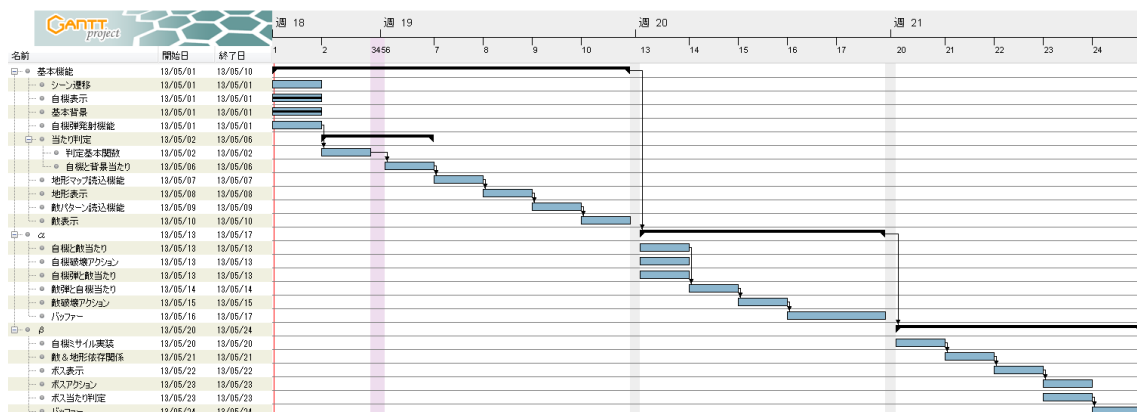
Google 大先生に『ガントチャート』でお伺いするとたくさん出てくると思います。こういうプロジェクト管理のツールは手を変え品を変え、ソフトウェア開発の歴史とともにありますが、『これでプロジェクトのデスマーチがなくなりました決定版』などという物はできていません。

プロジェクトは基本的には遅れるものとおいておいたほうが良いでしょう。しかし現実というのは非情なもので、プロジェクトの『ケツ』と『顧客が本当に必要なもの』はまず揺るがないものです。

つまり、最終成果物は遅れるわけには行かないし顧客の要求も満たさないと結局報酬は得られないわけです。

でも、開発中というのは一種の興奮状態でなかなかそれに気づけません。そういうもんです。ガントチャートってのは、そういうのを所謂『見える化』するツールだと思っておきましょう。

幾つかツールがありますが、僕が使っているツールは GanttProject とってこんな感じ



この使い方のコツは上記のようにいきなり細かいスケジュールを書いていくのではなく、最初は雑に割っていく感じで作りましょう。まだ細かいところまで決めてないでしょ？このガントチャート上では、それぞれの仕様を『タスク』と呼びます。

#### タスクの追加

新規タスクは右クリック→新規タスクで追加するか、Ctrl+Tで追加できます。

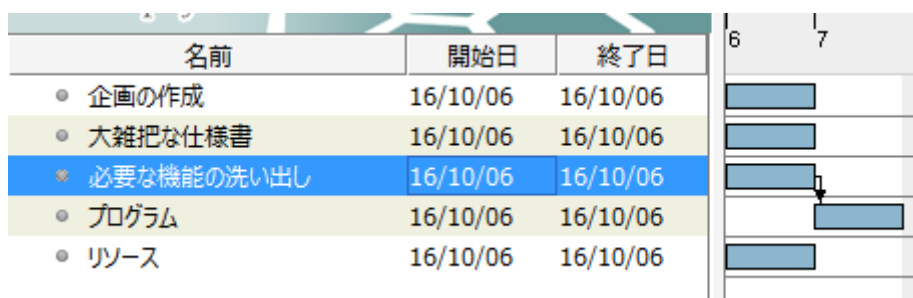
そうすると、このようにどんどん追加していくことができます。



#### タスクの依存関係

そして例えばプログラムなんかは『必要な機能』が分かってないと作れるものではないので、依存関係を作成します。

これは簡単に、前のタスクから次のタスクへドラッグアンドドロップすると線が伸びてこうなります。

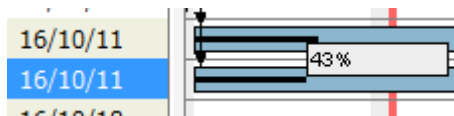


#### タスクの進捗

タスクは進捗を記録することができます。

それぞれのタスクのバーの左端からドラッグすると%マークが出てきて、数値が表示されま

すので、どのへんまで進捗したかを記録して行ってください。



こんな感じでパーセント表示されると思います。この現在の黒いリバーが、本日付より手前に来れば予定より『遅れている』という事になります。

#### 休日の設定

休日の設定はデフォルトでは設定されていません。土日くらいしか休みが設定されていませんが、当然、祝日やらあるわけですが、それは自分で設定しなければいけません。

プロジェクト→プロジェクトのプロパティ→プロジェクトのカレンダー→追加で追加していきます。ちなみに One-Off は一回こっきりの休み、Recurring は繰り返し休みのことです。

ちなみに、ここで一個一個やってくるのが面倒な場合には、gan ファイルをテキストファイルで開いて、中身が XML でできてますので、HOLIDAY の部分を追加してください。

#### ユースケース

<https://ja.wikipedia.org/wiki/%E3%83%A6%E3%83%BC%E3%82%B9%E3%82%B1%E3%83%BC%E3%82%B9>

このような定義がありますが、ここまでやらなくてもいいです。ちょっと世界が違う感じなので…ただ『ユースケースにおけるシナリオ』というのは大事なことです。

『シナリオ』っていうと物語の世界観だのそういうのを思い浮かべると思いますが、そうではなく、ユーザーがそのアプリケーションに対して、何を期待し、どのような行動を取ることが予測されるのか、どれくらいのパターンがあるのか、それに対してアプリケーションがどのようなアクションをして、それに対してまたユーザーがどのようなアクションを取るのかということを記述していくものです。

ここがしっかりしていないとゲームとして破綻したり、クソゲーになったりします。

ちなみに Wikipedia では

各ユースケースには少なくとも 1 つの基本シナリオあるいは典型的なイベントの流れが存在する。基本イベント経路は例えば以下のようなステップとして描かれる：

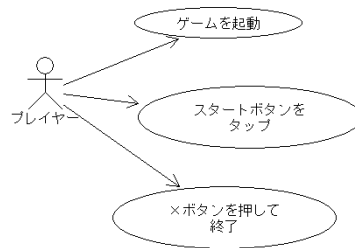
1. システムはユーザーに[ログオン](#)を促す。
2. ユーザーは名前とパスワードを入力する。
3. システムはそのログイン情報を確認する。

4. システムはシステムにログを残す。

... などなど。

と書かれています。

このようなものを『シナリオ』といいます。で、ユースケース図ってのがあるんですけど



図自体はどうでもいいんですけど、このようにユーザーの行動をきっちりしてないと、アプリケーションの仕様に漏れが発生したりします。