

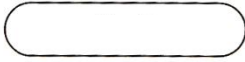
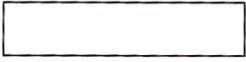

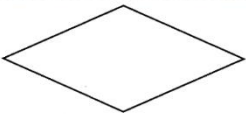
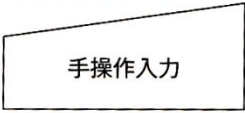



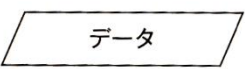

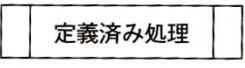
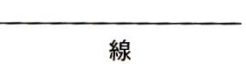
この時間のねらい：① 流れ図の作成を行うことができる。

② 自分達の作成するものの流れ図を作成する。

① \_\_\_\_\_ ( \_\_\_\_\_ ) とは

機械が処理を行うためには、どのような手順で行うのが大事になります。その手順のことを \_\_\_\_\_ と言います。また、人が処理をするときには、機械のように \_\_\_\_\_ ではありません。基本的には人為的なものが絡むため、順番どおりではないことが多いです。このような処理を \_\_\_\_\_ などと呼びます。これらは対義語とされるものです。(覚えておくと自慢できます。)

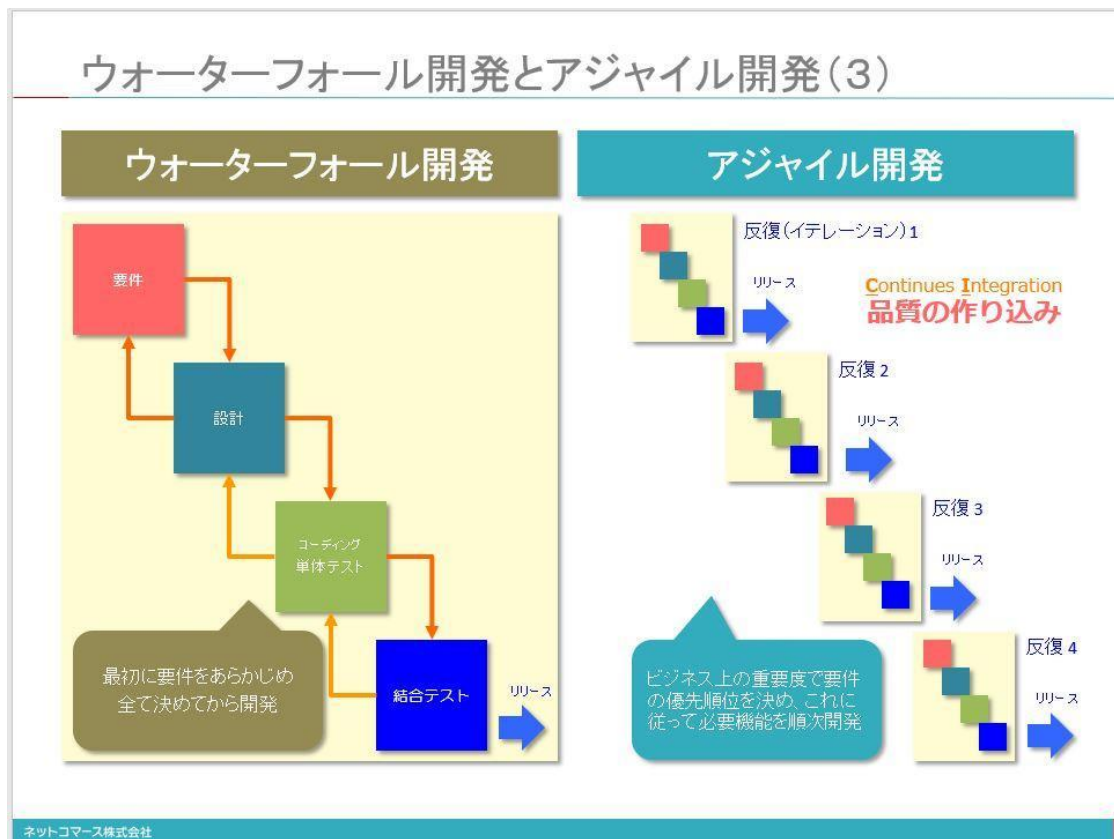
これを視覚的にわかりやすくするために、 \_\_\_\_\_ ( \_\_\_\_\_ ) というものを書きます。ちょっとしたお絵かきをするだけなのでそんなに苦ではないはずです。以下の記号を使用して書いていきます。

記号	意味	記号	意味
	プログラムの開始、終了または外部からの入口出口をあらわす。		演算や代入などあらゆる種類の処理をあらわす。
	初期値の設定など準備の処理をあらわす。		条件による比較、判断と処理の分岐をあらわす。
	キーボードからの入力をあらわす。		くりかえし処理のはじまりをあらわす。(ループ始端)
	ディスプレイ画面への出力をあらわす。		くりかえし処理の終わりをあらわす。(ループ終端)
	データの入力または出力をあらわす。		流れ図のつながりをあらわす。
	別の場所で定義された処理をあらわす。		データや制御の流れをあらわす。流れの向きを明示するときは矢印にする。

以上の記号を使って流れ図を書いていきます。

## 番外編) 開発モデルについて

皆さんには、開発を少しずつ始めてもらっています。基本的に皆さんにやってもらっている開発方式は\_\_\_\_\_と呼ばれるものになります。このほかに\_\_\_\_\_という方式もあります。それぞれの方式は以下のような流れで行われるものです。



### ウォーターフォール型開発モデル

#### ◆メリット

- ・開発中に混乱しづらい
- ・大規模な開発に強い
- ・無駄な時間がない

#### ◆デメリット

- ・アジャイル型のメリットがない

### アジャイル型開発モデル

#### ◆メリット

- ・途中で開発内容の変更がしやすい
- ・比較的早めに完成させられる
- ・依頼者の要望に応えやすい

#### ◆デメリット

- ・ウォーターフォール型のメリットがない

これ以外にも\_\_\_\_\_というモデルもあります。