

SOFTWARE ENGINEERING 2

STATISCHE ANALYSEN

- Untersuchung des Datenflusses auf Anomalien
- Zustand einer Variablen

u: undefiniert

d: definiert (d.h. Wert wurde zugewiesen)

r: referenziert (Wert der Variablen wird verwendet)

- Mögliche Anomalien

ur-Anomalie

du-Anomalie

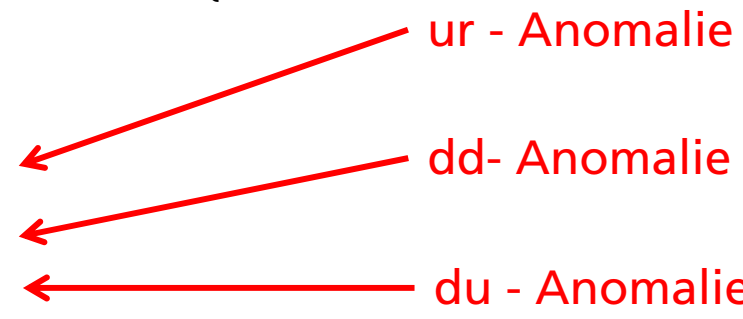
dd-Anomalie

```
void swap (int& min, int& max){  
    int tmp;  
    if(min > max){  
        max = tmp;  
        max = min;  
        tmp = min;  
    }  
}
```

ur - Anomalie

dd- Anomalie

du - Anomalie



Zyklomatische Komplexität – Beispiel

```
if (SourcePosition->isOccupied())
{
    Player *selectedPlayer=SourcePosition->getToken()->getTeam()->getPlayer();
    if (activePlayer == selectedPlayer)
    {
        bool successfulMove=SourcePosition->getToken()->move(TargetPosition);
        // on successfulMove, change Players alternately
        if (successfulMove)
        {
            // check if active player has won
            string sPlayerName=activePlayer->getName();

            if (activePlayer->hasWon())
            {
                stringstream sstream;
                string sMessage="Player ";
                sstream << sPlayerName << " wins the game!";

                mMessage = sMessage + sstream.str();
                setFinished();
                cout << mMessage << endl;
                continue; // the loop
            }
            if (sPlayerName == "White")
                setActivePlayer("Black");
            else
                setActivePlayer("White");
        }
        else
        { // Move not possible, the reason is in mMessage
        }
    }
    else // token is not of the active player
    {
        mMessage="Please choose a token of the active player. Try again." ;
    }
}
else // .. not occupied
{
    mMessage="Please choose a source position with a token on it. Try again." ;
}
```